



UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

**Departament de Teoria del Senyal
i Comunicacions**

Discovering Causality

A Degree Thesis
Submitted to the Faculty of the
Escola Tècnica d'Enginyeria de Telecomunicació de
Barcelona
Universitat Politècnica de Catalunya
by
Adrià Iglesias Rosich

In partial fulfilment
of the requirements for the degree in
Telecommunications Technologies and Services Engineering

Advisor: Jaume Riba Segarra

Barcelona, May 2018

Abstract

This thesis aims to study the field of signal processing applied to the detection of common causes in processes apparently from different origins. This will be done using adaptive filtering, more specifically three different adaptive algorithms have been selected that will be studied in depth and different simulations will be executed to evaluate their behavior. The field of correlation detection is applicable to many fields, from telecommunications, to economics, to neurology.

Resum

En aquesta tesi es pretén estudiar el camp del processat de senyal aplicat a la detecció de causes comunes en processos aparentment d'orígens diferents. Per a això es farà ús del filtrat adaptatiu, més concretament s'han seleccionat tres algorismes adaptatius diferents que s'estudiaran en profunditat i s'executaren diferents simulacions per avaluar el seu comportament. El camp de la detecció de correlació és aplicable a molts camps, des de telecomunicacions, fins economia, passant per neurologia.

Resumen

En esta tesis se pretende estudiar el campo del procesado de señal aplicado a la detección de causas comunes en procesos aparentemente de orígenes distintos. Para ello se hará uso del filtrado adaptativos, más concretamente se han seleccionado tres algoritmos adaptativos distintos que se estudiarán en profundidad y se ejecutaran distintas simulaciones para evaluar su comportamiento. El campo de la detección de correlación es aplicable a muchos campos, desde telecomunicaciones, hasta economía, pasando por neurología.

*Dedicado a mi familia que
me ha dado todo su apoyo
en este largo camino.*

Historial de Revisión y Aprobación

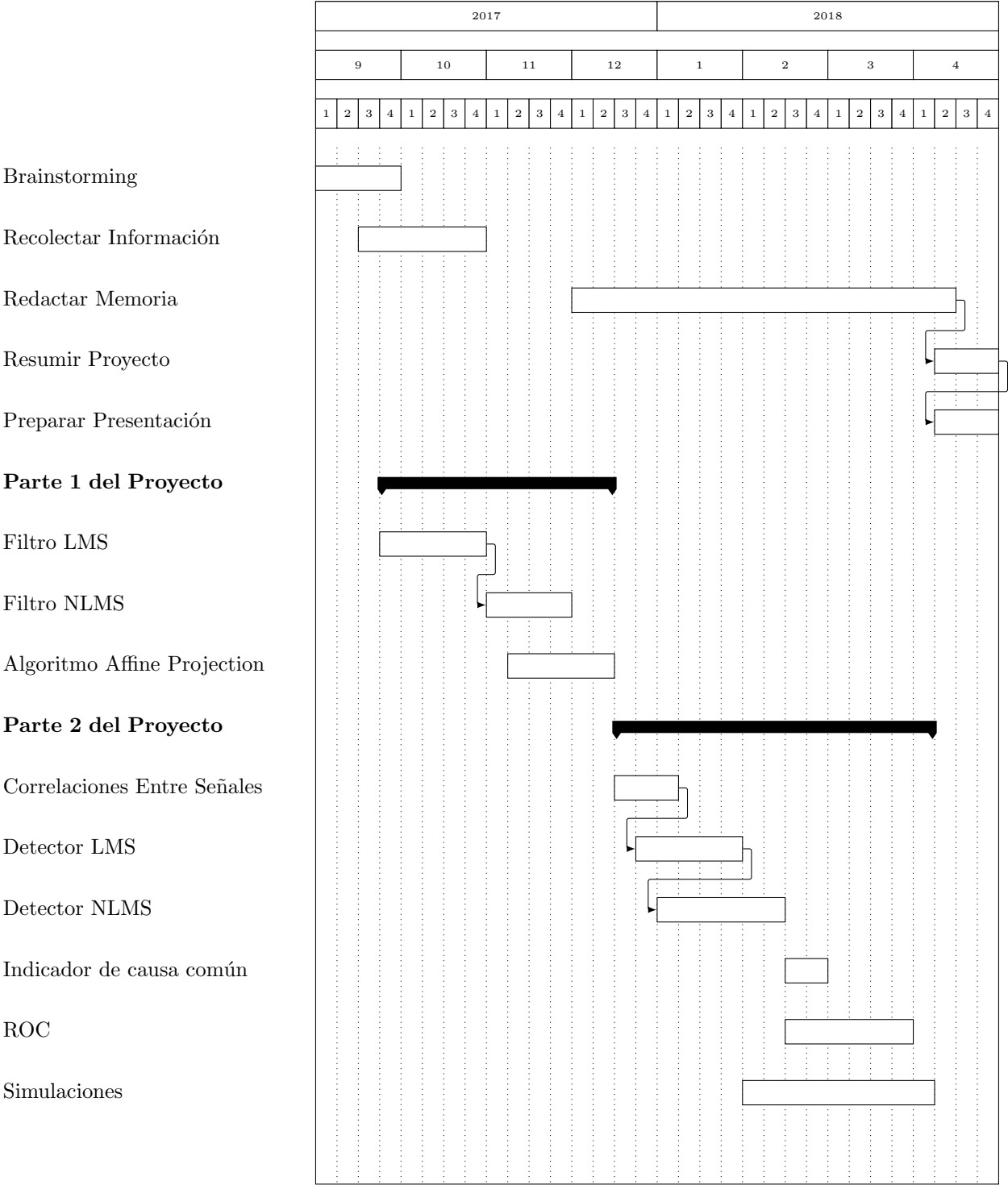
Revision	Date	Purpose
0	05/05/2018	Document creation
1	10/05/2018	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Adrià Iglesias Rosich	airadria15@gmail.com
Jaume Riba Segarra	jaume.riba@upc.edu

Written by: Adrià Iglesias Rosich		Reviewed and approved by: Jaume Riba Segarra	
Date	05/05/2018	Date	10/05/2018
Name	Adrià Iglesias Rosich	Name	Jaume Riba Segarra
Position	Project Author	Position	Project Supervisor

Diagrama de Gantt



Índice general

1. Motivaciones del poyecto	12
1.1. Resumen y organización del trabajo	12
2. Introducción	13
2.1. Conceptos esenciales	13
2.1.1. Notación vectorial	13
2.1.2. Operadores básicos	14
3. Filtrado adaptativo	15
3.1. Método del gradiente	15
3.1.1. Algoritmo del gradiente	15
3.2. Least-Mean-Square	16
3.2.1. Algoritmo LMS	16
3.3. Simulaciones LMS	16
3.3.1. Variando el tamaño del filtro real	17
3.3.2. Variando el valor de μ	17
3.3.3. Efecto de la dimensión de los filtros en la simulación	18
3.4. Normalized-Least-Mean-Square	20
3.4.1. Algortimo NLMS	20
3.5. Simulaciones NLMS	20
3.5.1. Variando la longitud del filtro real	20
3.5.2. Variando el valor de α	21
3.5.3. Sobredimensionamineto del los filtros	22
3.6. Affine Projection Algorithms	22
3.7. Simulaciones Affine Projection Algorithms	24
4. Detección de correlación	25
4.1. Canonical Correlation Analysis	25
4.1.1. Definición del análisis	25
4.2. Detección con Filtros Adaptativos	26
4.2.1. Algoritmo de Detección	26
4.2.2. Adaptación del Algoritmo de detección para detector Affine Projection	27
4.2.3. Indicador de Causa Comuna	28
4.3. Característica Operativa del Receptor	28
4.3.1. Área Bajo la Curva	29
4.4. Simulaciones	30
4.4.1. Detector LMS	30
4.4.2. Detector NLMS	33
4.4.3. Detector Affine Projection	36
5. Conclusiones	40
A. Identificador de Sistemas mediante LMS	43
B. Identificador de Sistemas mediante NLMS	44
C. Identificador de Sistemas mediante Affine Projection	45
D. Detector de Correlación LMS	46

E. Detector de Correlación NLMS	47
F. Detector de Correlación Affine Projection	48
G. Indicador de Causa Comuna	49

Índice de figuras

3.1. Filtro LMS configurado como identificador de sistemas	16
3.2. Error Cuadrático Medio a la salida del filtro LMS de 500 coeficientes	17
3.3. (a) Coeficientes del filtro real y del filtro LMS y (b) el error cuadrático medio a la salida del filtro LMS de 10 coeficientes	17
3.4. (a) Error cuadrático medio a la salida con μ igual a $\frac{4}{P_x}$ (b) Error cuadrático medio a la salida con μ igual a $\frac{8}{P_x}$ (c) Error cuadrático medio a la salida con μ igual a $\frac{16}{P_x}$	18
3.5. Error cuadrático medio resultante de la simulación con una variable stepsize sobredimensionada	18
3.6. (a) Evolución de los coeficientes, (b) Coeficientes del filtro real y del LMS y (c) Error cuadrático medio a la salida	19
3.7. (a) Evolución de los coeficientes, (b) Coeficientes del filtro real y del LMS y (c) Error cuadrático medio a la salida	19
3.8. Error Cuadrático Medio a la salida del filtro NLMS de 500 coeficientes	20
3.9. (a) Coeficientes del filtro real y del filtro NLMS y (b) el error cuadrático medio a la salida del filtro NLMS de 10 coeficientes	21
3.10. (a) Error en la simulación con α igual a 0.05 (b) Error en la simulación con α igual a 0.5	21
3.11. (a) Evolución de los coeficientes con el filtro adaptativo sobredimensionado, (b) Coeficientes del filtro real e identificado y (c) el error cuadrático medio resultante a la simulación	22
3.12. (a) Evolución de los coeficientes con el filtro real sobredimensionado, (b) Coeficientes del filtro real e identificado y (c) el error cuadrático medio resultante a la simulación	22
3.13. Diagrama de bloques de un filtro Affine Projection	23
3.14. Error Cuadrático Medio a la salida del filtro Affine Projection de 500 coeficientes	24
3.15. Error Cuadrático Medio a la salida del filtro Affine Projection de 10 coeficientes	24
4.1. Error Cuadrático Medio resultante de la detección con dos señales incorrelacionadas y con dos correlacionadas	27
4.2. Error Cuadrático Medio resultante de la detección con dos señales incorrelacionadas y con dos correlacionadas haciendo uso del indicador	28
4.3. Ejemplo de curvas Característica Operativa del Receptor con distintos detectores	29
4.4. Característica Operativa del Receptor del detector LMS sin indicador de causa comuna	31
4.5. Error Cuadrático Medio a la salida del detector para detección con señales correlacionadas y incorrelacionadas	31
4.6. Característica Operativa del Receptor del detector LMS con indicador de causa comuna	32
4.7. Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas	33
4.8. Característica Operativa del Receptor del detector NLMS sin indicador de causa comuna	34
4.9. Error Cuadrático Medio a la salida del detector para detección con señales correlacionadas e incorrelacionadas	34
4.10. Característica Operativa del Receptor del detector NLMS con indicador de causa comuna	35
4.11. Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas	36
4.12. Característica Operativa del Receptor del detector AP sin indicador de causa comuna	37
4.13. Error Cuadrático Medio a la salida del detector AP para detección con señales correlacionadas e incorrelacionadas	37
4.14. Característica Operativa del Receptor del detector AP con indicador de causa comuna	38
4.15. Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas	39
5.1. Error Cuadrático medio para cada algoritmo	41
5.2. Coeficientes de los filtros	41
5.3. (a) Error Cuadrático Medio sin indicador y (b) Error Cuadrático Medio con indicador	42

5.4. Curva Operativa del Receptor para cada detector	42
--	----

Índice de cuadros

4.1. Clasificación según el criterio de Swets para valores del AUC	29
4.2. Parámetros iniciales para la detección LMS sin indicador de causa comuna	30
4.3. Resultados para las distintos valores de la variable stepsize del detector LMS sin indicador de causa comuna	31
4.4. Parámetros iniciales para la detección LMS con indicador de causa comuna	32
4.5. Resultados para las distintos valores de la variable stepsize del detector LMS con indicador de causa comuna	32
4.6. Parámetros iniciales para la detección NLMS sin indicador de causa comuna	33
4.7. Resultados para las distintos valores de la variable α del detector NLMS sin indicador de causa comuna	34
4.8. Parámetros iniciales para la detección NLMS con indicador de causa comuna	35
4.9. Resultados para las distintos valores de la variable α del detector NLMS sin indicador de causa comuna	35
4.10. Parámetros iniciales para la detección AP sin indicador de causa comuna	36
4.11. Resultados para las distintos valores de la variable μ del detector AP sin indicador de causa comuna	37
4.12. Parámetros iniciales para la detección AP con indicador de causa comuna	38
4.13. Resultados para las distintos valores de la variable μ del detector AP con indicador de causa comuna	38

Capítulo 1

Motivaciones del proyecto

La primera asignatura de comunicaciones que cursé en el grado despertó en mi una inquietud frente a la rama de señales y su procesamiento. Desde entonces que tenía claro que debía hacer un proyecto relacionado con ello, para entender más profundamente los conceptos aprendidos a lo largo del grado. Por otro lado siempre me he sentido atraído por el mundo de la economía, en concreto por la bolsa. Tras pasar horas pensando como podía juntar estos dos temas, decidí que mi proyecto se basaría en determinar si algunas bajadas o subidas de empresas suponían bajadas o subidas de otras, en la bolsa.

Puesto que era un proyecto muy ambicioso, y más para alguien que desconoce por completo el sector bursátil, decidí, con la ayuda de mi tutor, centrarme exclusivamente en la detección de señales, que ya es un tema muy denso.

1.1. Resumen y organización del trabajo

La estructura de este estudio está dividida en tres bloques. El primer bloque, está formado por el capítulo 1 y 2 donde se exponen las motivaciones del trabajo y el objetivo que se tiene en mente. El segundo bloque, formado por los capítulos 2 y 3 se va a exponer todos los conceptos necesarios para la correcta comprensión de los resultados, además de mostrar distintos ejemplos necesarios para interiorizar los conceptos teóricos. En este bloque se va a definir la metodología de trabajo y las herramientas necesarias para poder entender al completo el segundo bloque. Además de las herramientas básicas, también estarán explicados e implementados una serie de filtros adaptativos que van a ser de gran ayuda para el correcto desarrollo del proyecto.

El tercer bloque, formado por los capítulos 4 y 5, se profundiza en el estudio de correlaciones, dependencia estadística entre señales, en el que se explican distintos métodos de detección y decisión. En el tercer bloque también se hallan las conclusiones del proyecto, donde se recolectan los resultados más trascendentales y se extraen ideas cruciales, y por último las perspectivas de futuro del proyecto.

La estructura más detallada del proyecto, capítulo a capítulo, es la siguiente:

- *Capítulo 2 “Introducción”*: En este capítulo se hace una exposición de conceptos aprendidos durante el grado que pueden ser útiles para el proyecto pero sobretodo que han motivado a llevarlo a cabo.
- *Capítulo 3 “Filtrado adaptativo”*: El eje principal de este capítulo van a ser los filtros adaptativos, se explicaran los antecedentes a estos sistemas, la necesidad de su creación, los principios matemáticos necesarios para poder entenderlos y se desarrollaran diferentes filtros para así poder entender los siguientes capítulos. En este capítulo también se exponen resultados del uso de los filtros adaptativos para la identificación de sistemas.
- *Capítulo 4 “Detección de correlación”*: El capítulo en cuestión, va a ser el más trascendental en el proyecto, en el se va a diseñar distintos sistemas detectores de señales correlacionas con y sin filtros adaptativos. Además también se expone un indicador que ayudará a determinar si se trata o no de señales correlacionas y por último propone un método para la evaluación del detector propuesto. A lo largo del capítulo se analizaran los resultados y a continuación se entenderá la dependencia estadística entre diferentes señales. Todo lo mencionado ira de la mano de gráficos que ilustrarán lo explicado y todos los resultados.
- *Capítulo 5 “Las Conclusiones y proyección de futuro”*: El último capítulo estará destinado a la exposición de conclusiones, mediante un resumen de los resultados obtenidos, su trascendencia y también una parte crítica donde se proponen aspectos de mejora además de exponer posibles continuaciones de este proyecto.

Capítulo 2

Introducción

Desde hace ya más de medio siglo que el procesamiento de señal es un campo que está en continuo desarrollo. Cada vez con más frecuencia aparecen nuevos algoritmos mejores que los anteriores, con menor coste computacional, menor distorsión, mayor SNR, etcétera. Cuando hablamos de filtrado adaptativo uno de los pioneros fue el profesor Bernard Widrow, quien desarrolló el algoritmo adaptativo más conocido y uno de los primeros, se trata del algoritmo Least Mean Squares (*LMS*). Dicho algoritmo fue publicado alrededor de los años 60 y hoy en día se sigue usando en infinidad de aplicaciones, debido a su simplicidad.

Los sectores donde podemos aplicar el procesamiento de señal son muchos, y es que en ocasiones se tiende a pensar que las telecomunicaciones es una ciencia cerrada donde no se puede extrapolar sus usos a otras materias. En la actualidad podemos encontrar procesamiento de señal en estudios de neurociencia, cardiología, economía y muchos más. En este trabajo se va a estudiar qué resultados se obtienen a la salida de un sistema, diseñado para detectar si dos o más señales están o no correlacionadas, cuando le introducimos diferentes entradas. Esto es útil en muchas ocasiones, por ejemplo ver si en bolsa existen cambios de tendencias de unas empresas dependiendo de las tendencias de otras, para ver interconexiones neuronales, etcétera.

La hipótesis inicial del proyecto es que si se va a poder encontrar correlaciones entre diferentes señales aplicando el criterio de detección.

2.1. Conceptos esenciales

Para la comprensión total de la notación usada en este proyecto es necesario comprender correctamente esta sección. A continuación se exponen conceptos relacionados con señales discretas, usados a lo largo del grado y de vital importancia.

2.1.1. Notación vectorial

Es importante definir una notación vectorial que ayude a facilitar los cálculos. La notación vectorial es de gran ayuda ya que existen multitud de programas que gestionan matrices de un modo muy ágil y rápido.

Run-time vector

El run-time vector de una señal es el vector formado por la muestra actual y las $M - 1$ muestras anteriores, y se define así:

$$\mathbf{x}[n] = \begin{bmatrix} x[n] \\ x[n-1] \\ \vdots \\ x[n-M+1] \end{bmatrix} \quad (2.1)$$

2.1.2. Operadores básicos

Convolución discreta

Se define $x[n]$ como una secuencia discreta a la entrada de un filtro lineal e invariante con respuesta impulsional $h^*[n]$, la salida de dicho sistema se obtiene mediante la convolución discreta:

$$y[n] = \sum_{m=-\infty}^{\infty} h^*[m]x[n-m] \quad (2.2)$$

El proyecto se centrará en el caso de filtros FIR causales, con lo que el sumatorio de la convolución dispondrá de un número finito de terminos, M coeficientes:

$$y[n] = \sum_{m=0}^{M-1} h^*[m]x[n-m] \quad (2.3)$$

De modo que se podrá definir la salida del filtro del siguiente modo:

$$y[n] = \mathbf{h}^H \mathbf{x}[n] \quad (2.4)$$

donde $\mathbf{x}[n]$ es el run-time vector y

$$\mathbf{h} = \begin{bmatrix} h[0] \\ h[1] \\ \vdots \\ h[M-1] \end{bmatrix} \quad (2.5)$$

Esperanza estadística

La esperanza de una variable aleatoria X , se define como la media de dicha variable aleatoria X y se calcula del siguiente modo:

$$E[X] = \int_{-\infty}^{\infty} x f(x) dx \quad (2.6)$$

Correlación estadística

Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra y se define tal que:

$$R_x[m] = E[x[n+m]x^*[n]] \quad (2.7)$$

Capítulo 3

Filtrado adaptativo

En el estudio de la econometría, se da una gran trascendencia a la estadística de las señales financieras pertenecientes a cada empresa listada en el mercado bursátil. Si se observa dicho problema con perspectiva, es fácil darse cuenta que no deja ser procesado de señal, entonces podemos extrapolar los conocimientos adquiridos en telecomunicaciones para aplicarlos en econometría, ya que seguimos estando en un escenario de procesamiento de señal. Entonces al tratarse de un problema donde desconocemos las características de las estadísticas de las señales, debemos hacer uso de los filtros adaptativos. Los filtros adaptativos permiten trabajar con señales de estadística desconocida gracias a que sus coeficientes son actualizados iteración tras iteración.

3.1. Método del gradiente

Antes de profundizar en el filtrado adaptativo, es importante introducir los métodos iterativos, en concreto, el método del gradiente.

3.1.1. Algoritmo del gradiente

El método en cuestión consiste en obtener el MSE, definido como:

$$\xi = E[(d(n) - \mathbf{h}^H \mathbf{x}[n])(d^*[n] - \mathbf{x}^H[n] \mathbf{h})] \quad (3.1)$$

De una forma iterativa, con ello evitamos complejidad matemática derivada del cálculo de la inversa de la matriz de autocorrelación. El cálculo de inversas requiere un coste computacional alto, además de requerir precisión elevada del resultado y además es posible que la matriz no tenga inversa.

El concepto en el que está basado el método, consiste en partir de un valor inicial del filtro (h_0) y actualizar su valor en dirección opuesta al MSE. La ecuación de actualización es la siguiente:

$$\mathbf{h}_{k+1} = \mathbf{h}_k - \mu \left. \frac{\partial \xi}{\partial \mathbf{h}^*} \right|_{\mathbf{h}=\mathbf{h}_k} \quad (3.2)$$

Como se puede observar en la ecuación, aparece la constante μ no definida aún. Dicha constante se conoce como *step size*, es de crucial importancia que esté bien definida, pues depende de ella el salto que se realice en cada iteración. Una mala elección de μ podría causar un sistema muy lento y aún peor, un sistema inestable. Volviendo a la ecuación 3.2, se observa que con el método *steepest descent*, se desciende lentamente por en dirección perpendicular a las curvas de nivel del MSE hasta llegar a la solución de Wiener.

El gradiente también podría obtenerse tal que:

$$\left. \frac{\partial \xi}{\partial \mathbf{h}^*} \right|_{\mathbf{h}=\mathbf{h}_k} = E[-\mathbf{x}[n](d^*[n] - \mathbf{x}^H[n] \mathbf{h})] = \mathbf{R}_x \mathbf{h}_k - \mathbf{r}_{dx} \quad (3.3)$$

De modo que concluimos que el algoritmo final es:

$$\mathbf{h}_{k+1} = \mathbf{h}_k - \mu(\mathbf{R}_x \mathbf{h}_k - \mathbf{r}_{dx}) \quad (3.4)$$

El algoritmo realiza iteraciones hasta que finalmente los coeficientes se han desplazado, dependientes del error de \mathbf{R}_x , \mathbf{h}_k y \mathbf{r}_{dx} , hasta llegar a la convergencia, es decir, cuando $(\mathbf{R}_x \mathbf{h}_k - \mathbf{r}_{dx}) = 0$.

3.2. Least-Mean-Square

El algoritmo Least-Mean-Square (*LMS*) [1], fue desarrollado por Widrow y Hoff en 1960, se trata de un algoritmo ampliamente usado, debido a su simplicidad, coste computacional, eficiencia y buen comportamiento ante gran variedad de condiciones. El algoritmo en cuestión usa un método iterativo que actualiza sus coeficientes en cada realización del proceso.

La decisión de elegir un método iterativo como es el algoritmo LMS, es debido a que con ello evitamos realizar la inversa matricial que se trata de una operación de un gran coste computacional, además el algoritmo Least-Mean-Square ofrece una gran sensibilidad a la hora de identificar sistemas, siempre que esté bien configurado.

3.2.1. Algoritmo LMS

Como se ha visto en la sección 3.1, el método del gradiente debe conocer la autocorrelación estadística de los datos y correlación estadística cruzada entre datos y referencia, esto hace que se trate de un método iterativo y no adaptativo. Una alternativa adaptativa a dicho método sería el algoritmo *Least-Mean-Square*, que realiza una estimación de la información a partir de los datos. Para el caso del algoritmo LMS, se usa una sola iteración por muestra, dando lugar a:

$$\mathbf{h}[n+1] = \mathbf{h}[n] - \mu(\mathbf{x}[n]\mathbf{x}^H[n]\mathbf{h}[n] - d^*[n]\mathbf{x}[n]) \quad (3.5)$$

Arreglando la 3.5, obtendremos el algoritmo Least-Mean-Square, desarrollado por Widrow y Hoff:

$$\mathbf{h}[n+1] = \mathbf{h}[n] - \mu\mathbf{x}[n]\epsilon^*[n] \quad (3.6)$$

donde la señal error está definida por:

$$\epsilon[n] = d[n] - \mathbf{h}^H[n]\mathbf{x}[n] \quad (3.7)$$

La siguiente ilustración muestra un diagrama de bloques del algoritmo LMS:

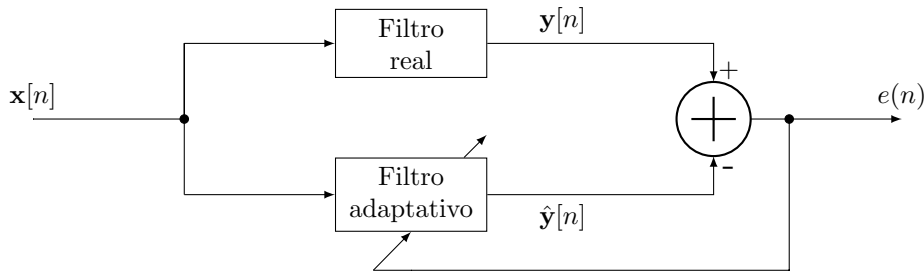


Figura 3.1: Filtro LMS configurado como identificador de sistemas

Es importante entender por qué la estimación de la estadística es fiable, esto se debe a que el filtro se encuentra configurado como lazo cerrado, es decir, como se halla realimentado, éste tiene memoria de las muestras anteriores y consigue que la estadística sea buena.

El filtro LMS ofrece la posibilidad de ser configurado de diferentes maneras con objetivos distintos, para el desarrollo del trabajo nos centraremos en la configuración de la figura 3.1, se trata de un identificador de sistemas. La configuración como identificador de sistemas es capaz de identificar un filtro desconocido solo con la señal de entrada y la de salida del filtro real a identificar. En la subsección 3.3 se evaluará el identificador de sistemas y se verá cómo actúa ante diferentes escenarios con el fin de ver sus virtudes y sus defectos.

3.3. Simulaciones LMS

A continuación se llevarán a cabo una serie de simulaciones del filtro LMS configurado como identificador de sistemas, para verificar todo lo explicado en la sección 3.2 y de ese modo, que se podrán extraer conclusiones con fundamentos. Las simulaciones se realizarán variando distintos parámetros que ayudarán a ver y entender los puntos fuertes y las debilidades del filtro LMS.

En el apéndice A se encuentra el código Matlab usado en las siguientes simulaciones.

3.3.1. Variando el tamaño del filtro real

La primera pregunta que a cualquiera se le pasaría por la cabeza en el momento de analizar y comprobar el modo de actuar de un filtro sería:

¿Qué sucedería si aumentase el tamaño del filtro? ¿Afectaría eso a su velocidad de convergencia?

A priori, se podría decir que no afectaría. A continuación se comprobará si la hipótesis planteada se cumple, o en su defecto, si se debería reformular.

Simulación con 500 muestras

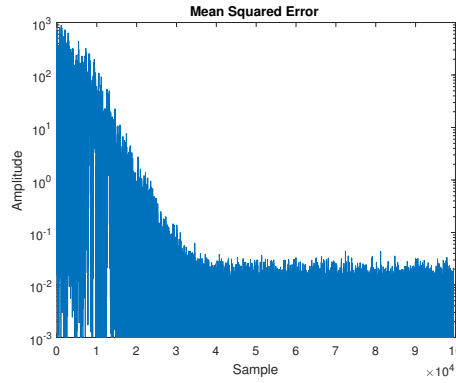


Figura 3.2: Error Cuadrático Medio a la salida del filtro LMS de 500 coeficientes

En la figura 3.2 se muestra el Error Cuadrático Medio de una simulación del filtro LMS con un filtro a identificar de 500 muestras. A partir de 3.5×10^4 muestras se puede considerar que el filtro ha sido identificado, pues el MSE se estabiliza y su valor es muy pequeño.

Simulación con 10 muestras

Con el objetivo de contrastar los resultados, se procederá a ejecutar el mismo identificador de sistemas pero esta vez se hará con un filtro real de 10 muestras, de modo que se verificará la hipótesis.

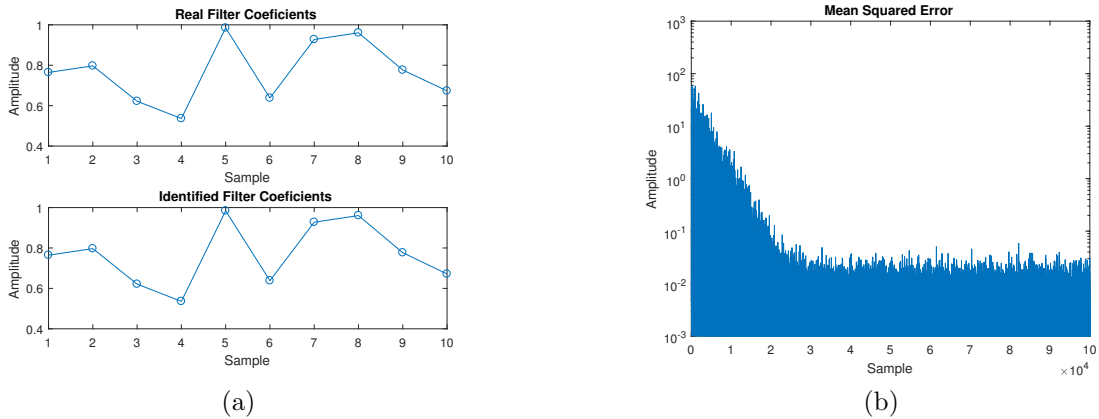


Figura 3.3: (a) Coeficientes del filtro real y del filtro LMS y (b) el error cuadrático medio a la salida del filtro LMS de 10 coeficientes

A simple vista se observa que el error de la figura 3.3 (b) es muy similar al de la figura 3.2 (b), de modo que podemos dar por verificada la hipótesis, afirmando con convicción que la velocidad de convergencia del filtro LMS no depende de la longitud del filtro real, a diferencia de otros algoritmos.

3.3.2. Variando el valor de μ

Si se ha entendido el significado de la variable *stepsize* (μ), explicado en secciones anteriores, se puede deducir que se trata de una variable de crucial importancia en el algoritmo, esto es así porque la variable μ

define el tamaño del salto que realiza el algoritmo en cada iteración, se puede definir como la tasa de aprendizaje. Entonces, se puede intuir que como menor sea el tamaño de μ , menor será el salto y, en consecuencia, menor será la velocidad de convergencia. Se podría caer en la tentación de afirmar que como mayor sea μ , más rápido, convergerá, de ser así nos estaríamos equivocando pues si se sobrepasa el tamaño óptimo, la velocidad disminuirá. En caso de asignar un valor excesivamente grande a μ , el sistema podría convertirse en inestable.

Con el objetivo de comparar los resultados obtenidos con distintos valores de la tasa de aprendizaje, se han realizado tres simulaciones con la misma señal de datos e idéntica configuración, excepto por el hecho de haber cambiado el valor de μ para cada una de ellas.

Comparación con μ de diferentes valores

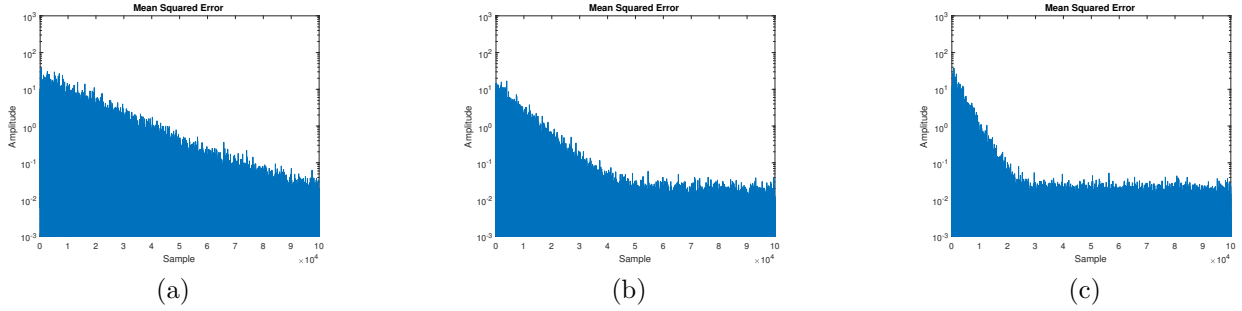


Figura 3.4: (a) Error cuadrático medio a la salida con μ igual a $\frac{4}{P_x}$ (b) Error cuadrático medio a la salida con μ igual a $\frac{8}{P_x}$ (c) Error cuadrático medio a la salida con μ igual a $\frac{16}{P_x}$

Es fácil observar que en el caso donde la variable *stepsize* tiene un valor de $\frac{4}{P_x}$, figura 3.4 (a), donde P_x es la potencia de la señal de entrada el error requiere aproximadamente 10×10^4 muestras para identificar el sistema. En cambio, en el caso donde la variable *stepsize* es igual a $\frac{8}{P_x}$, figura 3.4 (b), se observa que en el tiempo de convergencia es inferior a la mitad del anterior, y por último en la figura 3.4 (c), cuando $\mu = \frac{16}{P_x}$, el tiempo de convergencia es aún menor, lo que nos llevar a concluir que efectivamente la magnitud de la tasa de aprendizaje es inversamente proporcional al tiempo de convergencia, hay que recordar que siempre se debe cumplir $\mu \leq 1$.

Sobredimensionamiento de μ

A continuación se comprobará qué sucede si asignamos a la variable *stepsize* un valor mayor al máximo que el sistema puede permitir para no convertirse en inestable. Para ello no es necesario que $\mu > 1$.

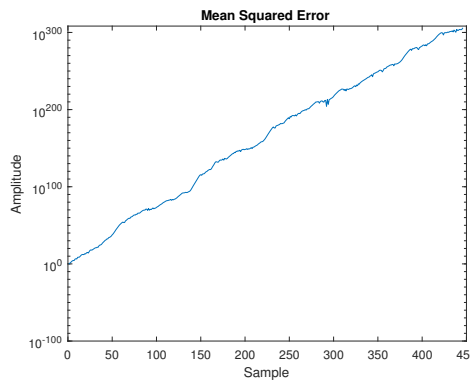


Figura 3.5: Error cuadrático medio resultante de la simulación con una variable *stepsize* sobredimensionada

Como se puede observar en la figura 3.5 el error cuadrático medio tiende a infinito, esto se debe a que el sistema es inestable. El sistema se vuelve inestable cuando el salto que realiza el algoritmo, en la iteración, se sale del paraboloide elíptico, entonces el sistema se descontrola y nunca llegará al resultado deseado.

3.3.3. Efecto de la dimensión de los filtros en la simulación

Otra situación a tener en cuenta cuando valoramos el comportamiento del filtro, sería cuando las dimensiones del filtro real h y del filtro adaptativo no son coincidentes. A priori se podría decir que el sobredimensionamiento

del filtro adaptativo no supondría ningún problema a cuando se quiere identificar el filtro real, en cambio, en el caso contrario cuando se sobredimensiona el filtro real, el resultado nunca podría llegar a ser el deseado.

Sobredimensionamiento del filtro adaptativo

Para comprobar qué sucede cuando se sobredimensiona el filtro adaptativo, se empezará por la simulación con un filtro adaptativo con un número de coeficientes superior al del filtro real que se quiere identificar. Como se ha dicho en la introducción de la subsección 3.3.3, se toma como hipótesis que la mala asignación de la longitud del filtro adaptativo no supodría una adversidad para encontrar el filtro.

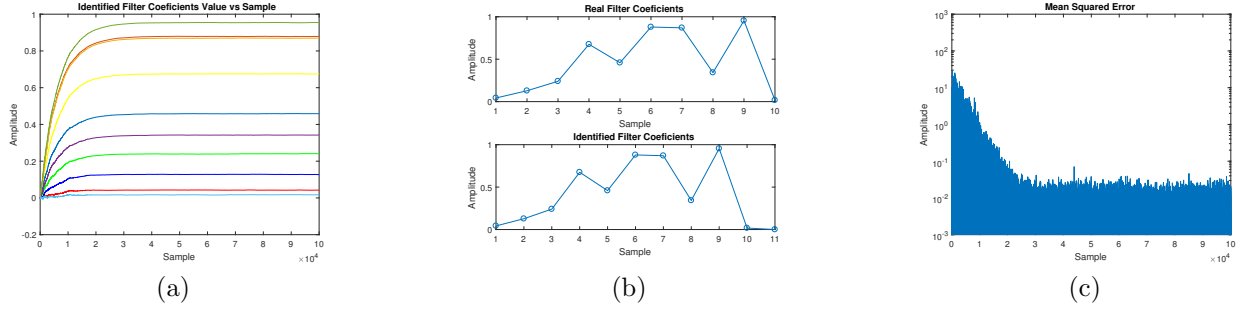


Figura 3.6: (a) Evolución de los coeficientes, (b) Coeficientes del filtro real y del LMS y (c) Error cuadrático medio a la salida

Como se ha realizado la simulación con la misma *stepsize* que en la sección 3.3.1, se procederá a contrastar ambos resultados.

Mediante el error, definido en la figura 3.6 (c), queda demostrado que el sistema ha sido identificado por completo, de no ser así, no se habría obtenido un Error Cuadrático Medio de la misma magnitud que en las simulaciones anteriores. Ahora bien, si se observa el gráfico donde aparecen representados los coeficientes, figura 3.6 (b), el coeficiente número 11 del filtro adaptativo se mantiene a cero, debido a que este no existe en el filtro real. Se podría decir que el sobredimensionado del filtro adaptativo es estéril frente a la identificación del sistema.

Sobredimensionamiento del filtro real

En este caso se comprobará qué sucede cuando el filtro sobredimensionado es el real. Como se ha hecho en el caso anterior donde el sobredimensionado era el filtro adaptativo, los resultados se compararán con la subsección 3.3.1.

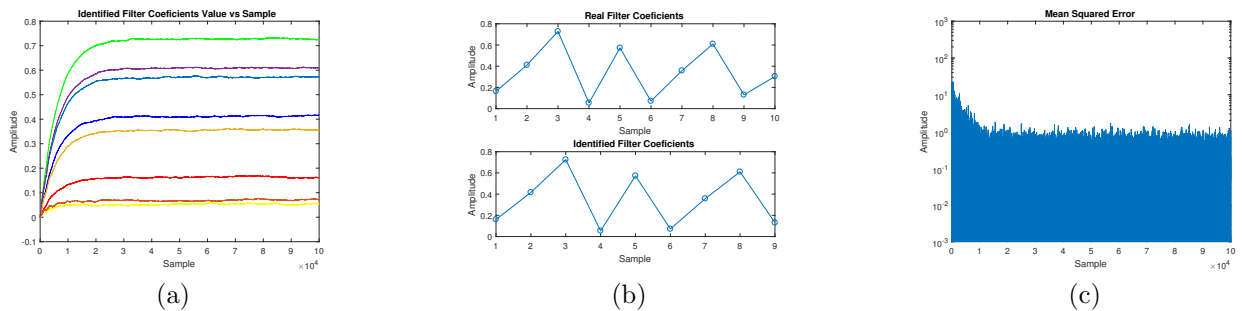


Figura 3.7: (a) Evolución de los coeficientes, (b) Coeficientes del filtro real y del LMS y (c) Error cuadrático medio a la salida

En el gráfico donde se representa el Error Cuadrático Medio, figura 3.7 (c), se observa que este nunca al nivel de las simulaciones anteriores, donde no existía sobredimensionado, independientemente del número de iteraciones que realice el algoritmo. Esto es consecuencia de la falta de un coeficiente del filtro adaptativo, para que el error llegue a ser cero, es necesario que la distancia entre dos filtros también sea nula, en el caso actual, como las dimensiones de los filtros son diferentes la distancia nunca podrá llegar a ser cero y por tanto el error tampoco llegará a ser nulo.

3.4. Normalized-Least-Mean-Square

El algoritmo Normalized-Least-Mean-Square (NLMS), [2] es una variación del algoritmo LMS, sección 3.2. El principal inconveniente del algoritmo LMS es que es sensible a la escala de la entrada, $\mathbf{x}[n]$. Esto supone que sea difícil, sino imposible, elegir la variable *stepsize*, μ , que garantice la estabilidad del sistema. El filtro NLMS es una variante del algoritmo LMS que resuelve este problema mediante la normalización del μ con la potencia de la entrada [3].

Como alternativa al algoritmo LMS, con el algoritmo NLMS evitamos la principal dificultad del anterior, pues al disponer de una variable *stepsize* variante, en cada iteración, los resultados se obtendrán con mayor rapidez y se evitará obtener un resultado inestable a la salida del filtro.

3.4.1. Algoritmo NLMS

En el caso del Algoritmo NLMS, como no se conoce la potencia de los datos una de las opciones para fijar la variables *stepsize*, podría ser la siguiente:

$$\mu[n] = \frac{2\alpha}{\delta + \|\mathbf{x}[n]\|^2} \quad (3.8)$$

En la ecuación anterior, la idea es usar una estimación instantánea de \mathbf{R}_x como $\mathbf{x}[n]\mathbf{x}^H[n]$. En la ecuación anterior, se define $\delta > 0$, como un valor pequeño para evitar problemas numéricos si la estimación instantánea, $\|\mathbf{x}[n]\|^2$, es muy pequeña.

En definitiva, el algoritmo NLMS se define igual que el algoritmo LMS pero con la diferencia de que la variable μ varía en cada iteración.

La ecuación de actualización resultante sería:

$$\mathbf{h}[n+1] = \mathbf{h}[n] - \frac{2\alpha}{\delta + \|\mathbf{x}[n]\|^2} \mathbf{x}[n](d[n] - \mathbf{h}^H[n]\mathbf{x}[n])^* \quad 0 < \alpha < 1 \quad (3.9)$$

3.5. Simulaciones NLMS

En esta sección se llevarán a cabo las simulaciones de las mismas situaciones planteadas en la sección 3.3, pero esta vez se hará una breve comparación entre ambos algoritmos, LMS i NLMS. En el apéndice B se halla el código usado en las siguientes simulaciones del filtro NLMS.

3.5.1. Variando la longitud del filtro real

Igual que se ha hecho en el capítulo anterior, la primera pregunta que uno se plantea cuando quiere comprobar el modo de actuar de un filtro sería:

¿Qué sucedería si se aumentase el tamaño del filtro? ¿Afectaría eso a la velocidad de convergencia del mismo?

A priori se podría decir que no afectaría, igual que en el algoritmo LMS, pero realmente no es así. A continuación se verá que aplicando el algoritmo LNLMS, la longitud del filtro real es un factor a tener en cuenta.

Simulación con 500 muestras

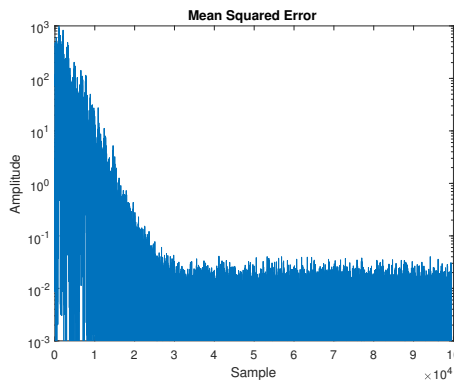


Figura 3.8: Error Cuadrático Medio a la salida del filtro NLMS de 500 coeficientes

Como se puede apreciar en el gráfico 3.8 (b), el error empieza a ser despreciable a partir de la muestra 2.5×10^4 , por tanto se puede considerar que el sistema ha sido identificado. Comparando la figura 3.8 con la figura 3.2 se puede apreciar que el tiempo de convergencia es inferior, del orden de 1,5 veces más rápido que para el caso del LMS.

Simulación con 10 muestras

Para poder contrastar los resultados, se procederá a ejecutar el mismo identificador de sistemas pero esta vez se hará con un filtro real de 10 muestras, de modo que se verificará la hipótesis.

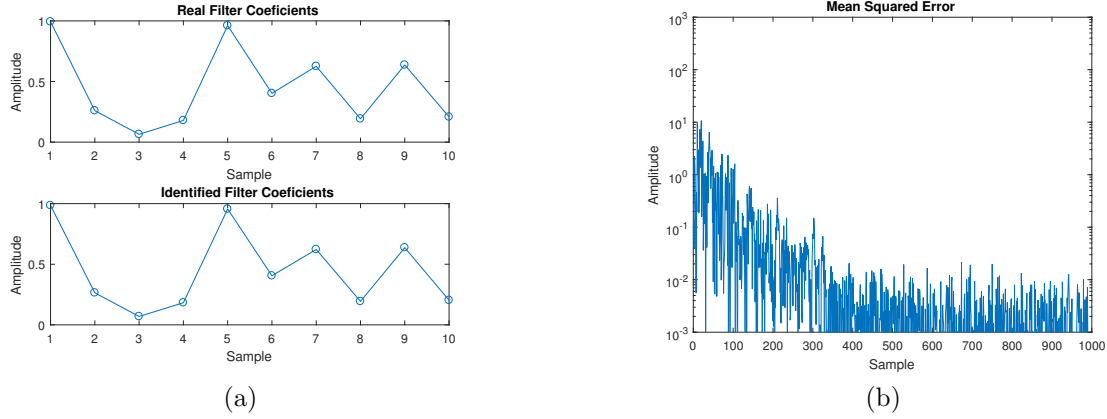


Figura 3.9: (a) Coeficientes del filtro real y del filtro NLMS y (b) el error cuadrático medio a la salida del filtro NLMS de 10 coeficientes

A simple vista se observa que el error de la figura 3.9 es muy distante al de la figura 3.8, mientras que el primero únicamente tardaba 2.5×10^4 muestras a converger, en este caso el algoritmo resuelve el problema en 350 iteraciones, afirmando que la velocidad de convergencia del filtro NLMS depende de la longitud del filtro real.

3.5.2. Variando el valor de α

Si se observa detenidamente la definición matemática del algoritmo NLMS, uno se da cuenta que la variable α es de crucial importancia. Se intuye que como menor sea la variable α , mayor será el nombre de realizaciones para que el sistema conveja y a la inversa.

Por tal de ejecutar un gran número de realizaciones, se harán las mismas simulaciones que en la sección 3.5.1 y se contrastará debidamente los resultados.

Comparación con α de distintos valores

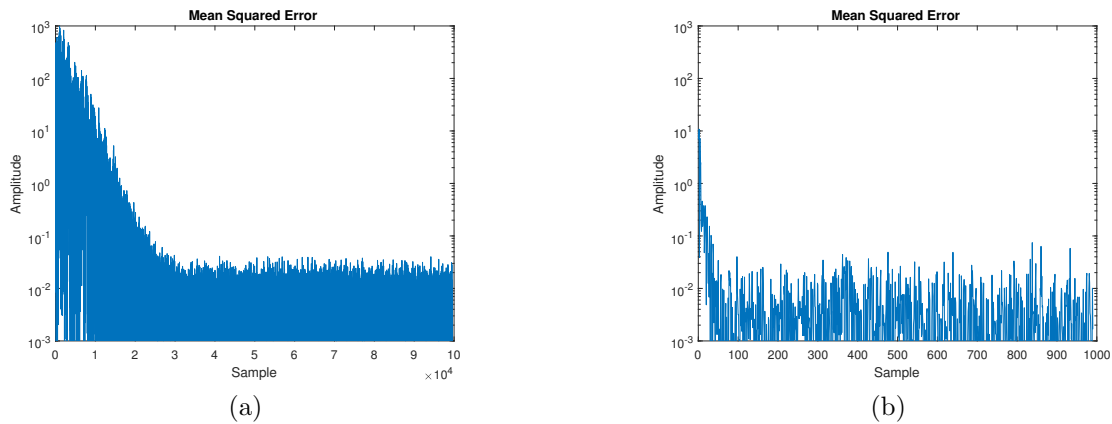


Figura 3.10: (a) Error en la simulación con α igual a 0.05 (b) Error en la simulación con α igual a 0.5

Después de realizar diversas simulaciones con diferentes valores de α , se puede afirmar que la mayor velocidad de convergencia se da cuando α es igual a 0.5. Cuando el valor de dicha variable α es menor que 0.5, entonces el

filtro es más lento a la hora de identificar el sistema, en cambio cuando es mayor sucede algo parecido a lo que sucedía en el filtro LMS con la variable μ superior al valor óptimo, los coeficientes del filtros en cada iteración sufren un cambio excesivo y eso hace que no se pueda llegar al valor óptimo, resultando un MSE superior a las otras simulaciones.

3.5.3. Sobredimensionamiento de los filtros

Filtro adaptativo sobredimensionado

Se empezará por simular el filtro adaptativo con un número de coeficientes superior al del filtro real que se quiere identificar. Como bien se ha dicho en la introducción de la subsección 3.3.3, se toma como hipótesis que la mala asignación de la longitud del filtro adaptativo no supondrá una adversidad para identificar el filtro.

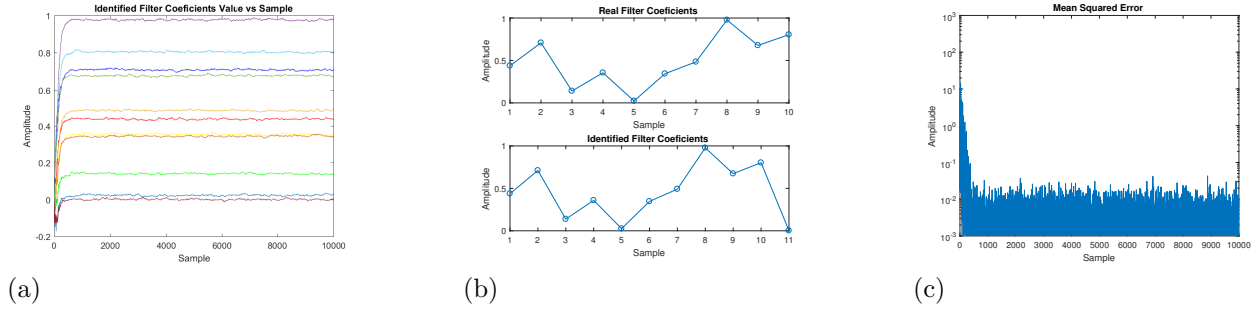


Figura 3.11: (a) Evolución de los coeficientes con el filtro adaptativo sobredimensionado, (b) Coeficientes del filtro real e identificado y (c) el error cuadrático medio resultante a la simulación

Se puede observar claramente que el comportamiento del algoritmo NLMS cuando se sobredimensiona el filtro adaptativo es muy similar al del algoritmo LMS, el coeficiente de más en el filtro adaptativo permanece a cero.

Filtro real sobredimensionado

En este caso se comprobará qué sucede cuando el filtro sobredimensionado es el real.

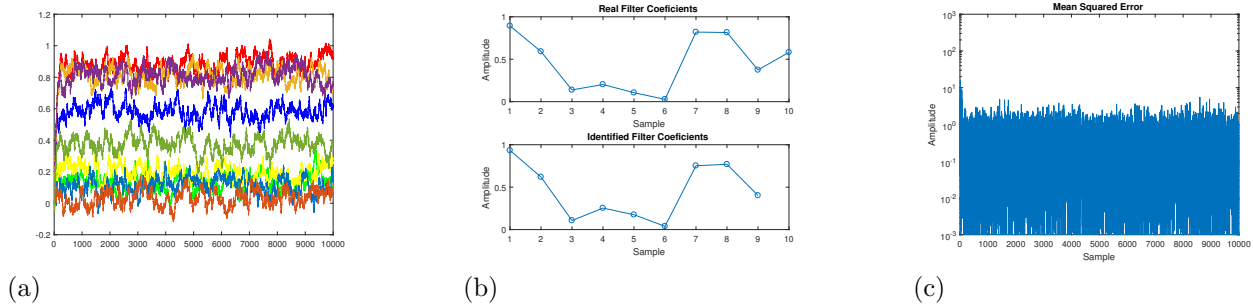


Figura 3.12: (a) Evolución de los coeficientes con el filtro real sobredimensionado, (b) Coeficientes del filtro real e identificado y (c) el error cuadrático medio resultante a la simulación

En la representación del Error Cuadrático Medio se observa claramente que este nunca llegará a cero, independientemente del número de iteraciones que realice el algoritmo. Eso es consecuencia de la falta de un coeficiente del filtro adaptativo. Por tal de que el error llegue a cero, sería necesario que la distancia entre los coeficientes del filtro fuese cero, pero como el filtro real es de mayor dimensión, esto nunca podría suceder. Además, debido a la diferencia entre ambas dimensiones, los coeficientes identificados disponen de un desajuste considerable.

3.6. Affine Projection Algorithms

Existen una gran variedad de filtros, algoritmos y métodos para lograr identificar sistemas conociendo su entrada y su correspondiente salida. Es interesante entender por qué anteriormente se ha hecho uso de los algoritmos least-mean-square y su variante normalizada, con ellos se evita realizar la inversa de la matriz de correlación de los datos. A continuación se presenta un algoritmo que permite obtener una mejora más que considerable en la identificación del sistema, sin repercutir en la sensibilidad del resultado, la decisión de elegir

este algoritmo es que nos ofrece una solución entre medio del algoritmo LMS y los algoritmos RLS (Recursive-Least-Squares algorithms) que permite mejorar la velocidad de detección de correlación de señales, el principal objetivo del proyecto, y conseguir la autovaloración de la constante de adaptación.

El algoritmo en cuestión pertenece a la familia de los algoritmos Affine Projection [4], dichos (*algoritmos AP*) agrupan una serie de algoritmos configurados para mejorar los resultados de otros algoritmos adaptativos, principalmente respecto a los algoritmos LMS, sobretodo cuando los datos están altamente correlacionados. Con el uso de un algoritmo AP se pretende conseguir un algoritmo eficiente y autoreconfigurable.

Los algoritmos AP pueden entenderse como una extensión del algoritmo NLMS [5]. Estos algoritmos se definen del siguiente modo:

$$\Delta \mathbf{w}_L[n] = \mathbf{w}_L[n] - \mathbf{w}_L[n-1] \quad (3.10)$$

Donde la ecuación 3.10 es la actualización de los L coeficientes del filtro adaptativo, en cada iteración del algoritmo.

Para desarrollar correctamente el algoritmo se debe minimizar la ecuación 3.11 con las N constantes obtenidas con la ecuación 3.12.

$$\|\Delta \mathbf{w}_L[n]\|^2 = \Delta \mathbf{w}_L^T[n] \Delta \mathbf{w}_L[n] \quad (3.11)$$

$$\mathbf{w}_L^T[n] \mathbf{x}_L[n-k] = d[n-k] \quad \text{para } k = 0, 1, \dots, N-1 \quad (3.12)$$

En la ecuación anterior aparece el vector $\mathbf{x}_L[n-k]$, dicho vector está compuesto por las L últimas muestras del vector de datos $\mathbf{x}[n]$ y $d[n]$ representa la señal deseada.

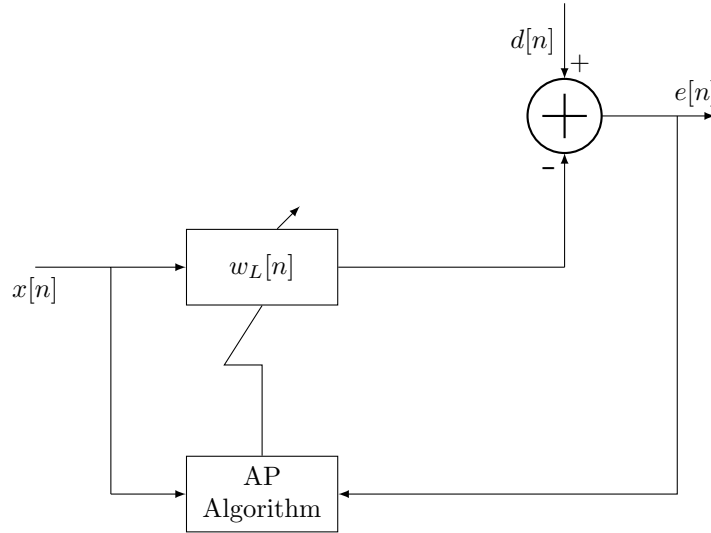


Figura 3.13: Diagrama de bloques de un filtro Affine Projection

La solución al problema reside en la ecuación de actualización del algoritmo AP

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}_N[n] \quad (3.13)$$

donde $\mathbf{A}[n]$ se define como

$$\mathbf{A}[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-1], \dots, \mathbf{x}_L[n-N+1])^T \quad (3.14)$$

y $\mathbf{e}_N[n]$ es un vector de dimensión $N \times 1$ tal que

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n] \mathbf{w}_L[n-1] \quad (3.15)$$

por último, $\mathbf{d}_N[n]$ representa el vector de la señal deseada de dimensión $N \times 1$

$$\mathbf{d}_N^T[n] = (d[n], d[n-1], \dots, d[n-N+1]) \quad (3.16)$$

3.7. Simulaciones Affine Projection Algorithms

En esta sección se procederá a comentar los resultados obtenidos para las simulaciones con el identificador de sistemas diseñado a partir del algoritmo Affine Projection. En el análisis se muestran dos simulaciones variando la longitud del filtro a identificar.

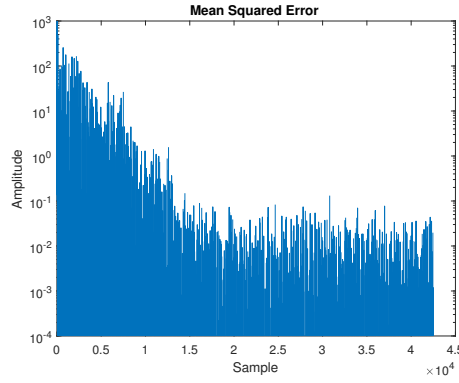


Figura 3.14: Error Cuadrático Medio a la salida del filtro Affine Projection de 500 coeficientes

En la simulación con 500 muestras para el algoritmo AP, deben transcurrir 1500 muestras para converger en el resultado. Si se compara con los filtros LMS y NLMS se observa que la velocidad de convergencia se multiplica por dos y se ratifica el motivo por el que se seleccionó el algoritmo AP, pues este al trabajar con matrices muestra una mejora en el tiempo de identificación.

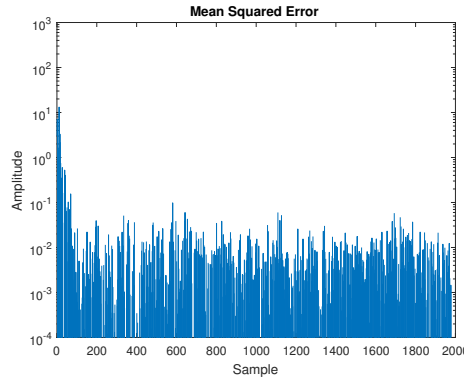


Figura 3.15: Error Cuadrático Medio a la salida del filtro Affine Projection de 10 coeficientes

Por otro lado, la figura 5.2 muestra otra simulación con 10 muestras y el resultado es el esperado, al reducir los coeficientes a identificar el tiempo de convergencia se reduce considerablemente. En esta simulación, la diferencia respecto a los algoritmos LMS y NLMS aún es mucho mayor siendo 500 veces más rápido que el LMS y 7 veces más rápido que el NLMS.

Capítulo 4

Detección de correlación

En este capítulo se pretende estudiar la dependencia estadística entre señales, para ello vamos a usar la correlación. En probabilidad y estadística, la correlación indica la fuerza y la dirección de una relación lineal y proporcionalidad entre dos variables aleatorias. Estudiando la dependencia estadística se conseguirá descubrir si los procesos aleatorios estudiados tienen una causa común. Esto puede ser de gran utilidad en el mundo de la inversión, donde suele haber correlaciones entre mercados y activos (bolsa, deuda, divisas, materias primas) que se mueven a unísono tanto a la alza como a la baja y esto puede simplificar la labor de los gestores de fondos de inversión, tanto a la hora de posicionarse como a la de cubrirse ante posibles caídas. Además de en los mercados de inversión, todo lo estudiado en este trabajo es extrapolable a otros ámbitos como la telecomunicaciones, medicina, ciencia, etcétera.

Para descubrir correlaciones entre señales existen varios métodos, pero siguiendo con la dinámica de esta tesis, se usarán los filtros adaptativos explicados en el capítulo anterior para descubrir si los procesos aleatorios tienen o no una misma causa común.

4.1. Canonical Correlation Analysis

Com bien se ha comentado en la introducción de este capítulo, se quiere detectar correlaciones entre dos señales, para ello introduciremos el método que da nombre a esta sección, Análisis de la Correlación Canónica (CCA) [6]. Desarrollado por Harold Hotelling en 1936, el método es una herramienta de estadística que mide la dependencia lineal entre dos vectores de datos, la idea que se presentara consiste en reformular el método mediante dos filtros LMS para descubrir si dos señales tienen una misma causa común. El análisis de la correlación canónica es usado en una variedad de campos como meteorología, economía, procesamiento de la señal y otros.

4.1.1. Definición del análisis

Supongamos $\mathbf{X}_1, \mathbf{X}_2$ dos matrices de datos de rango completo de dimensiones $N \times m_1$ y $N \times m_2$, respectivamente. El Análisis de la Correlación Canónica [7] se puede definir como el problema de encontrar los vectores \mathbf{w}_1 de dimensión $m_1 \times 1$ y \mathbf{w}_2 de dimensión $m_2 \times 1$, tal que las variables $\mathbf{y}_1 = \mathbf{X}_1 \mathbf{w}_1$ y $\mathbf{y}_2 = \mathbf{X}_2 \mathbf{w}_2$ se encuentran en correlación máxima. De modo que:

$$\arg \max_{\mathbf{w}_1, \mathbf{w}_2} \rho = \frac{\mathbf{y}_1^T \mathbf{y}_2}{\|\mathbf{y}_1\| \|\mathbf{y}_2\|} = \frac{\mathbf{w}_1^T \mathbf{R}_{12} \mathbf{w}_2}{\sqrt{\mathbf{w}_1^T \mathbf{R}_{11} \mathbf{w}_1 \mathbf{w}_2^T \mathbf{R}_{22} \mathbf{w}_2}} \quad (4.1)$$

Donde $\mathbf{R}_{ij} = \mathbf{X}_j^T \mathbf{X}_i$ es un estimador de la matriz de correlación. El problema de la ecuación anterior se puede reescribir considerandolo un problema de optimización de constantes tal que:

$$\arg \max_{\mathbf{w}_1, \mathbf{w}_2} \mathbf{w}_1^T \mathbf{R}_{12} \mathbf{w}_2 \quad \text{sujeto a} \quad \mathbf{w}_1^T \mathbf{R}_{11} \mathbf{w}_1 = \mathbf{w}_2^T \mathbf{R}_{22} \mathbf{w}_2 = 1 \quad (4.2)$$

La solución a la ecuación 4.1.1 se obtiene mediante el autovector asociado al mayor autovalor del siguiente problema generalizado de autovalores:

$$\begin{bmatrix} \mathbf{0} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{0} \end{bmatrix} \mathbf{v} = \rho \begin{bmatrix} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{0} & \mathbf{R}_{22} \end{bmatrix} \mathbf{v} \quad (4.3)$$

Donde ρ es la máxima correlación entre las dos variables \mathbf{w}_1 y \mathbf{w}_2 y $\mathbf{v} = [\mathbf{w}_1^T, \mathbf{w}_2^T]^T$ es el autovector.

4.2. Detección con Filtros Adaptativos

Existen muchos métodos para realizar la detección de correlaciones, uno de ellos es haciendo uso de los filtros adaptivos, estudiados en el capítulo 3, donde se exponían el filtro Least-Mean-Square, Normalized-Least-Mean-Square y el algoritmo Affine Projection.

4.2.1. Algoritmo de Detección

La idea en la que se basa el método de detección mediante filtros adaptativos consiste en partir de la hipótesis de que se dispone de dos procesos $x[n]$ e $y[n]$. El propósito es determinar si existe o no correlación entre ambos procesos, de no existir correlación alguna, entonces estaríamos hablando de procesos incorrelacionados, los cuales no comparten una causa común. Para ello, se va a intentar predecir una combinación lineal de las muestras del proceso $y[n]$ a partir de otra combinación lineal de las muestras de $x[n]$ y viceversa.

Debido a la reciprocidad del problema, el modo en que se procederá será simétrico, partiendo de los dos siguientes vectores columna de dimensión N :

$$\mathbf{x}[n] \quad \mathbf{y}[n] \quad (4.4)$$

Se generan los dos vectores columna, no nulos, de dimensión N que reproducen las combinaciones lineales de cada proceso.

$$a[n] = \mathbf{u}^T[n]\mathbf{x}[n] \quad (4.5)$$

$$b[n] = \mathbf{v}^T[n]\mathbf{y}[n] \quad (4.6)$$

A partir de los dos procesos 4.5 y 4.6 se obtendrá el proceso error definido tal que:

$$e[n] = a[n] - b[n] = \mathbf{u}^T[n]\mathbf{x}[n] - \mathbf{v}^T[n]\mathbf{y}[n] \quad (4.7)$$

Cabe destacar que para el problema planteado existe una solución trivial al algoritmo, un modo de hacer el error nulo a pesar de no existir correlación entre los procesos 4.4. Dicha solución es resultado de los vectores \mathbf{v} y \mathbf{u} sean nulos ($\mathbf{v} = \mathbf{u} = \mathbf{0}$), por ello es crucial evitar la solución trivial que nos impida medir la existencia de una razón común mediante el error.

Una vez definidos los procesos con que se procederá a trabajar, se define el cuadrado del error instantáneo y sus correspondientes gradientes para los vectores \mathbf{v} y \mathbf{u} :

$$e^2[n] = (\mathbf{u}^T[n]\mathbf{x}[n] - \mathbf{v}^T[n]\mathbf{y}[n])^2 \quad (4.8)$$

$$\frac{\partial}{\partial \mathbf{u}} e^2[n] = 2e[n]\mathbf{x}[n] \quad (4.9)$$

$$\frac{\partial}{\partial \mathbf{v}} e^2[n] = -2e[n]\mathbf{y}[n] \quad (4.10)$$

El algoritmo se basa en adaptar ambos combinadores en la dirección contraria al gradiente instantáneo con un factor de aprendizaje μ , es importante recordar que la variable μ será genérica para el algoritmo LMS y en el caso del NLMS será específica para cada proceso entrante, pero la ecuación de actualización los dos casos es la misma:

$$\mathbf{u}[n+1] = \mathbf{u}[n] - \mu e[n]\mathbf{x}[n] \quad (4.11)$$

$$\mathbf{v}[n+1] = \mathbf{v}[n] - \mu e[n]\mathbf{y}[n] \quad (4.12)$$

Como se ha mencionado anteriormente, el algoritmo no funcionará correctamente en vista a la tendencia a hacer decrecer las normas de \mathbf{u} y \mathbf{v} a fin de conseguir llevar el error asintóticamente a cero. Para evitar la tendencia natural al decaimiento hacia cero del algoritmo, este requiere ser estimulado minuciosamente para no desviarlo de su objetivo, medir la presencia de una causa común. La propuesta consiste en que el estímulo únicamente provoque un cambio de escala en el error, para ello es indispensable que la modificación afecte por igual a \mathbf{u} que a \mathbf{v} , en definitiva, se usará el mismo factor α para los dos. El error resultante de la estimulación es el siguiente:

$$(\alpha \mathbf{u}^T[n]\mathbf{x}[n] - \alpha \mathbf{v}^T[n]\mathbf{y}[n])^2 = \alpha^2 e^2[n] \quad (4.13)$$

En relación a la ecuación 4.13 obtenemos el algoritmo definido tal que:

$$\mathbf{u}' = \mathbf{u}[n] - \mu e[n]\mathbf{x}[n] \quad (4.14)$$

$$\mathbf{v}' = \mathbf{v}[n] - \mu e[n]\mathbf{y}[n] \quad (4.15)$$

$$\alpha = \frac{1}{\sqrt{\|\mathbf{u}'\|^2 + \|\mathbf{v}'\|^2}} \quad (4.16)$$

$$\mathbf{u}[n+1] = \alpha \mathbf{u}' \quad (4.17)$$

$$\mathbf{v}[n+1] = \alpha \mathbf{v}' \quad (4.18)$$

Tal y como se ha propuesto el algoritmo, se consigue que en cada iteración se cumpla la restricción que $\|\mathbf{u}[n]\|^2 + \|\mathbf{v}[n]\|^2 = 1$, evitando que el mismo decaiga a cero. Se observa que la condición anterior impide que ambos combinadores lineales sean simultáneamente nulos.

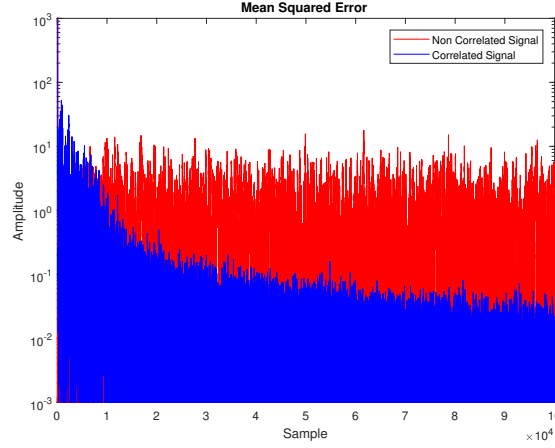


Figura 4.1: Error Cuadrático Medio resultante de la detección con dos señales incorrelacionadas y con dos correlacionadas

4.2.2. Adaptación del Algoritmo de detección para detector Affine Projection

Con el objetivo de aplicar el algoritmo Affine Projection a la detección de correlación y siguiendo en la línea del proyecto de aumentar la velocidad de convergencia, motivación principal del algoritmo AP, se debe adaptar el algoritmo de detección, usado para los detectores LMS y su correspondiente normalizado, el NLMS, al algoritmo Affine Projection.

Debido a la reciprocidad del problema, se procederá de forma simétrica, partiendo de las siguientes matrices, de dimensión $N \times L$:

$$\mathbf{A}[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-1], \dots, \mathbf{x}_L[n-N+1])^T \quad (4.19)$$

$$\mathbf{B}[n] = (\mathbf{y}_L[n], \mathbf{y}_L[n-1], \dots, \mathbf{y}_L[n-N+1])^T \quad (4.20)$$

A continuación se definen dos vectores columna, no nulos, de dimensión L que representan las combinaciones lineales de cada proceso:

$$\mathbf{a}[n] = \mathbf{A}[n] \mathbf{w}_u[n] \quad (4.21)$$

$$\mathbf{b}[n] = \mathbf{B}[n] \mathbf{w}_v[n] \quad (4.22)$$

A partir de los dos procesos 4.21 y 4.22 se obtendrá el proceso error definido tal que:

$$\mathbf{e}[n] = \mathbf{a}[n] - \mathbf{b}[n] = \mathbf{A}[n] \mathbf{w}_u[n] - \mathbf{B}[n] \mathbf{w}_v[n] \quad (4.23)$$

Siendo $w_u[n]$ y $w_v[n]$ los coeficientes del filtro Affine Projection A y B respectivamente, cuya ecuación de actualización es la siguiente:

$$\mathbf{w}_u[n+1] = \mathbf{w}_u[n] - \mu \mathbf{A}^T[n] (\mathbf{A}[n] \mathbf{A}^T[n])^{-1} \mathbf{e}[n] \quad (4.24)$$

$$\mathbf{w}_v[n+1] = \mathbf{w}_v[n] + \mu \mathbf{B}^T[n] (\mathbf{B}[n] \mathbf{B}^T[n])^{-1} \mathbf{e}[n] \quad (4.25)$$

Para evitar la solución trivial, ámbos vectores de coeficientes del filtro nulos, se define:

$$\alpha = \frac{1}{\sqrt{\|\mathbf{w}_u\|^2 + \|\mathbf{w}_v\|^2}} \quad (4.26)$$

$$\mathbf{w}_u[n+1] = \alpha \mathbf{w}_u \quad (4.27)$$

$$\mathbf{w}_v[n+1] = \alpha \mathbf{w}_v \quad (4.28)$$

4.2.3. Indicador de Causa Comuna

Una vez definido el algoritmo de detección mediante el filtrado adaptativo, con el objetivo de ser precisos, se debe definir un indicador en el cual se pueda apreciar claramente si los procesos son correlacionados o en su defecto incorrelacionados. Para ello se ha propuesto un índice basado en el error cuadrático, obtenido con el algoritmo de detección, definido tal que:

$$q[n+1] = (1 - \lambda)q[n] + \lambda e^2[n] \quad (4.29)$$

Donde $e[n]$ es el error resultante del algoritmo adaptativo desarrollado en la subsección 4.2.1 y λ controla la velocidad de decrecimiento del indicador. Como se ha definido el indicador $q[n]$, este tenderá a ser pequeño cuando ambos procesos provengan de filtrados distintos de un mismo proceso, y en cambio, tenderá a ser grande cuando los procesos estén generados a partir del filtraje de dos procesos independientes.

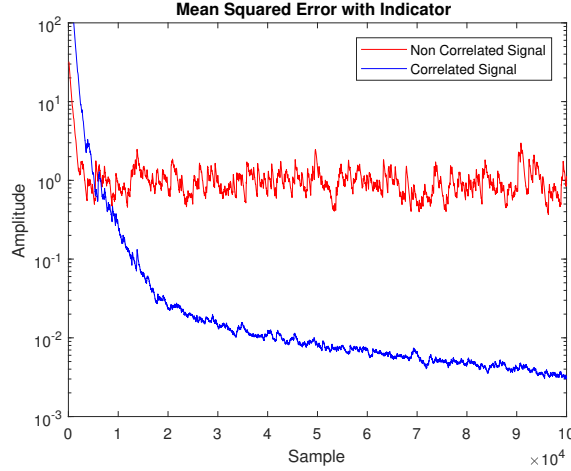


Figura 4.2: Error Cuadrático Medio resultante de la detección con dos señales incorrelacionadas y con dos correlacionadas haciendo uso del indicador

En los gráficos anteriores se puede apreciar que el indicador planteado puede resultar una buena herramienta para decidir si existe causa conjunta. Ya que este acaba con gran parte de los cambios repentinos de valor mostrando únicamente la tendencia de la recta.

4.3. Característica Operativa del Receptor

Una vez llegados a este punto del proyecto, es importante evaluar la fiabilidad del detector propuesto, pues carece de utilidad si su probabilidad de error es alta. En procesamiento de señal existe una herramienta llamada Característica Operativa del Receptor (ROC) [8], cuya razón de ser es evaluar la habilidad de decisión de un clasificador binario, como por ejemplo el propuesto en la sección 4.2.

Para explicar correctamente en que consiste la ROC, previamente es necesario hacer una breve exposición sobre teoría de detección.

Se dispone de una variable aleatoria con una función densidad de probabilidad que puede ser $N(0, \sigma^2)$ (Hipótesis H_0) o bien $N(A, \sigma^2)$ (Hipótesis H_1). Tras realizar una observación $x[0]$, la dificultad está en decidir si el resultado pertenece a la hipótesis H_0 o en su defecto a la H_1 . En resumen:

$$H_0 : x[0] = w[0] \quad (4.30)$$

$$H_1 : x[0] = A + w[0] \quad (4.31)$$

Se puede plantear un detector basado en un umbral γ , tal que si $x[0] > \gamma$ se trata de la hipótesis H_1 y si $x[0] < \gamma$ se trata de la H_0 .

Una vez definido el detector, pueden suceder dos tipos de error. El primero sucede cuando decidimos H_1 pero en realidad se dio H_0 y el segundo caso cuando decidimos H_0 pero en realidad se dio H_1 . En relación a los posibles resultados, se definen las siguientes probabilidades:

- Probabilidad de detección: $P(H_1; H_1) = P_D$
- Probabilidad de miss-detection: $P(H_0; H_1) = 1 - P_D = P_{MD}$
- Probabilidad de falsa alarma: $P(H_1; H_0) = P_{FA}$

En la mayoría de situaciones se presta un gran interés en conseguir una probabilidad de falsa alarma muy pequeña, puesto que las consecuencias de una falsa alarma pueden repercutir en grandes costes. Mediante la probabilidad de falsa alarma se impone un umbral de decisión que resultará en una probabilidad de detección. En conclusión, cuanto mayor sea la probabilidad de detección una vez fijado el umbral, mejor será el detector.

Volviendo al ejemplo anterior, en esta ocasión se dispone de N realizaciones independientes. Entonces, para cada valor fijado de probabilidad de falsa alarma, tendremos una determinada probabilidad de detección. Si la probabilidad de falsa alarma especificada es muy pequeña, ello obligará a un umbral muy elevado, con lo que la probabilidad de detección será también pequeña. Por otro lado, si se permite una probabilidad de falsa alarma más elevada, ello permite bajar el umbral y con ello aumentar la probabilidad de detección. El compromiso entre la probabilidad de falsa alarma y la probabilidad de detección queda bien representado mediante una curva ROC que representa la probabilidad de detección en función de la probabilidad de falsa alarma para un detector dado.

4.3.1. Área Bajo la Curva

Es interesante poder definir un parámetro bajo el que se mesure la capacidad que tiene el detector de discriminar resultados, el método elegido para ello es el Área Bajo la Curva (AUC) [9]. Con este parámetro estadístico se podrá comparar entre distintos detectores y decidir cuál de ellos proporciona más credibilidad.

El Área Bajo la Curva (AUC) se define tal que:

$$AUC = \int_0^1 ROC(x) dx \quad (4.32)$$

donde $ROC(x)$ es la función asociada a la curva ROC.

Los resultados obtenidos se pueden clasificar en tres distintos rangos definidos en el criterio de Swets:

Cuadro 4.1: Clasificación según el criterio de Swets para valores del AUC

Baja exactitud:	[0.5 , 0.7]
Útiles para algunos propósitos:	[0.7 , 0.9]
Exactitud alta:	[0.9 , 1]

Tal y como indica el cuadro 4.1 el criterio de Swets determina que un área inferior a 0'7 dispone de una baja capacidad discriminante, un área por debajo de 0'9 puede ser útil para algunas utilidades y las superiores a 0'9 se determina que disponen de una alta exactitud.

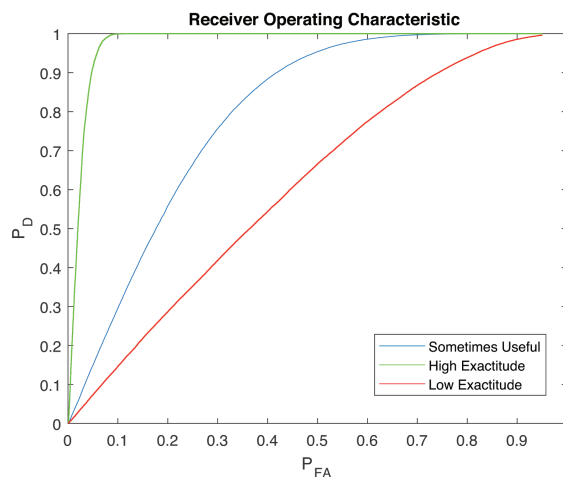


Figura 4.3: Ejemplo de curvas Característica Operativa del Receptor con distintos detectores

La función verde da figura 4.3 muestra un caso óptimo donde el área bajo la curva es $AUC = 0,9264$, en esta situación puesto que el área bajo la curva es próxima al máximo, se dispone de un detector con alta probabilidad de detección. Esta situación es la deseada ya que a dicho sistema se le podrá asignar un umbral mayor y los resultados seguirán siendo correctos, es decir el sistema tendrá una gran sensibilidad.

A pesar de desear un detector con alta exactitud, en muchas situaciones los sistemas no ofrecen tanta sensibilidad y disponen de un área bajo la curva en el intervalo $[0,7, 0,9]$, como muestra la función azul la figura 4.3, que se caracteriza por un $AUC = 0,7453$. Este sistema no ofrece tanta sensibilidad como en el mencionado anteriormente, pero la probabilidad de error de detección sigue siendo razonable.

Por último, la función roja de la figura 4.3 muestra el peor de los casos, donde el área bajo la curva $AUC = 0,5638$, dicho resultado es consecuencia de una probabilidad de error es muy elevada. Esta situación se debe evitar ya que los resultados carecen de utilidad para detectar correlaciones entre señales.

4.4. Simulaciones

Una vez expuesta toda la teoría, se presentarán las distintas simulaciones de los detectores realizados a lo largo del proyecto y se dará una visión crítica de ellos.

4.4.1. Detector LMS

Para elaborar el detector LMS, se han usado dos filtros LMS, definidos en la sección 3.2, configurándolos, como se expone en la subsección 4.2.1, de modo que a la salida del sistema se obtiene una señal error que será la clave para determinar si las señales entrantes están o no correlacionadas.

Con el objetivo de inspeccionar el comportamiento del sistema, se mostrarán sus ejecuciones en distintos escenarios y los resultados serán estudios de prestaciones promedio del sistema:

Detección sin indicador de causa comuna LMS

El primer escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante la señal error cuadrático medio a la salida del detector LMS.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.2: Parámetros iniciales para la detección LMS sin indicador de causa comuna

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Longitud señales de entrada:	100000 muestras

Los siguientes gráficos muestran el comportamiento del sistema de detección mediante filtros LMS cuando no se hace uso del indicador de causa comuna:

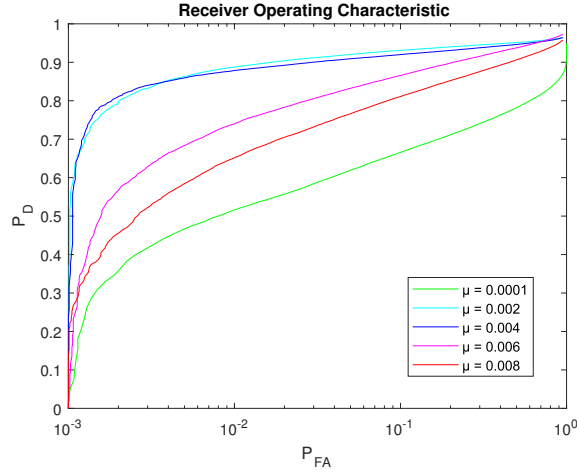


Figura 4.4: Característica Operativa del Receptor del detector LMS sin indicador de causa comuna

Cuadro 4.3: Resultados para los distintos valores de la variable stepsize del detector LMS sin indicador de causa comuna

Configuración	AUC
$\mu = 0,0001$	$AUC = 0,7679$
$\mu = 0,002$	$AUC = 0,8985$
$\mu = 0,004$	$AUC = 0,8932$
$\mu = 0,006$	$AUC = 0,8748$
$\mu = 0,008$	$AUC = 0,8409$

Después de las distintas simulaciones del sistema se puede observar como el comportamiento del sistema, en términos de probabilidad de detección, varía considerablemente para los distintos valores de la μ planteados en el experimento. Se puede verificar lo comentado anteriormente en relación a la inestabilidad del sistema, pues la velocidad de convergencia de este crece dependiendo de la tasa de aprendizaje, hasta su valor óptimo, y entonces empieza a decrecer. Al usar una variable stepsize mayor de lo deseado el sistema puede convertirse en inestable y con ello aparecer falsos positivos.

Para este sistema, la mejor realización sucede con una $\mu = 0,002$ cuando se obtiene una $AUC = 0,8985$. El resultado es muy próximo al de un filtro de con Alta Exactitud. Se puede observar que a partir del valor obtenido, si se sigue incrementándolo, empieza a disminuir el Área Bajo la Curva.

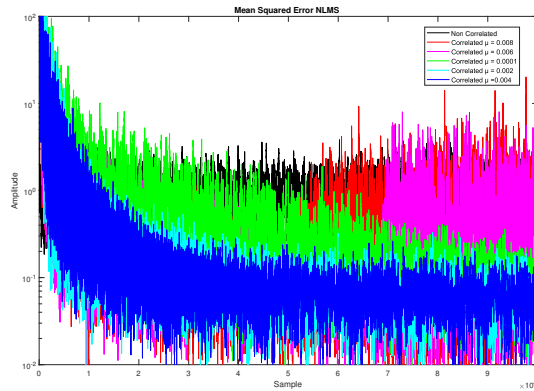


Figura 4.5: Error Cuadrático Medio a la salida del detector para detección con señales correlacionadas y incorrelacionadas

Se puede observar en la figura 4.5 como el MSE resultante de la detección de señales correlacionadas decrece, mientras el error resultante de la detección de señales incorrelacionadas se mantiene estable. Se puede ver como

para la simulación con mayores valores de tasa de aprendizaje, el sistema llega un punto que se vuelve inestable, tal y como se ha comentado.

Detección con indicador de causa comuna LMS

El segundo escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante el siguiente indicador de causa comuna:

$$q[n+1] = (1 - \lambda)q[n] + \lambda e^2[n] \quad (4.33)$$

Con la presencia del indicador se aspira a reducir la probabilidad de error de detección. Tal y como está definido, el indicador reduce los posibles picos que se producen en el error, y que podrían generar una falsa alarma.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.4: Parámetros iniciales para la detección LMS con indicador de causa comuna

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Longitud señales de entrada:	100000 muestras
Controlador velocidad decrecimiento del indicador (λ):	0.02

Los siguientes gráficos muestran el comportamiento del sistema de detección mediante filtros LMS cuando se hace uso del indicador de causa comuna:

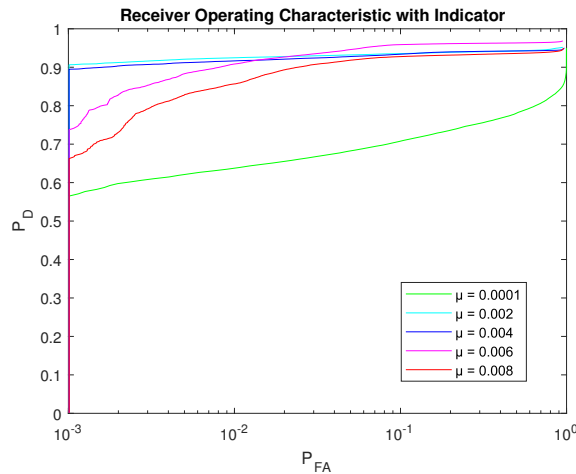


Figura 4.6: Característica Operativa del Receptor del detector LMS con indicador de causa comuna

Cuadro 4.5: Resultados para los distintos valores de la variable stepsize del detector LMS con indicador de causa comuna

Configuración	AUC
$\mu = 0,0001$	$AUC = 0,7745$
$\mu = 0,002$	$AUC = 0,8935$
$\mu = 0,004$	$AUC = 0,9120$
$\mu = 0,006$	$AUC = 0,9115$
$\mu = 0,008$	$AUC = 0,8947$

Los resultados de esta simulación del experimento no distan demasiado respecto al anterior, pues al tratarse del mismo sistema pero añadiendo un indicador de causa comuna, su comportamiento frente a cambios de variables es el mismo. En esta ocasión, el valor óptimo de la tasa de aprendizaje ha sido $\mu = 0,004$ con una Área Bajo la Curva $AUC = 0,9120$. Se puede apreciar una mejora de un 1,31 % del AUC obteniendo así un sistema con Alta Exactitud

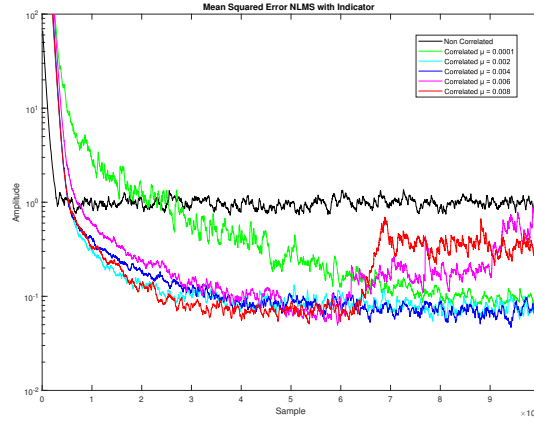


Figura 4.7: Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas

Como se puede apreciar en la figura 4.7, al aplicar el indicador, cuando ambas señales entrantes son correlacionadas, la señal que se obtiene a la salida del filtro carece de cambios bruscos de valor, se podría decir que es más estable. En cambio, cuando la señales entrantes no disponen de causa comuna, siguen existiendo cambios repentinos de valor.

Aplicando el indicador, como consigue reducir los cambios bruscos de valor en la señal resultante del sistema, se reduce la posibilidad de un falso negativo.

4.4.2. Detector NLMS

Para elaborar el detector NLMS, se han usado dos filtros NLMS, definidos en la sección 3.4, configurándolos, como se expone en la subsección 4.2.1, de modo que a la salida del sistema se obtiene una señal error que será la clave para determinar si las señales entrantes están o no correlacionadas.

Con el objetivo de inspeccionar el comportamiento del sistema, se mostrarán sus ejecuciones en distintos escenarios:

Detección sin indicador de causa comuna NLMS

El primer escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante la señal error a la salida del detector NLMS.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.6: Parámetros inicales para la detección NLMS sin indicador de causa comuna

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Longitud señales de entrada:	100000 muestras

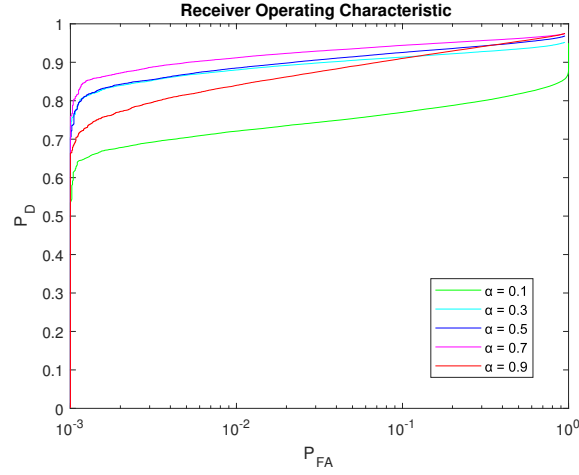


Figura 4.8: Característica Operativa del Receptor del detector NLMS sin indicador de causa comuna

Cuadro 4.7: Resultados para los distintos valores de la variable α del detector NLMS sin indicador de causa comuna

Configuración	AUC
$\alpha = 0,1$	$AUC = 0,8123$
$\alpha = 0,3$	$AUC = 0,8836$
$\alpha = 0,5$	$AUC = 0,8979$
$\alpha = 0,7$	$AUC = 0,9097$
$\alpha = 0,9$	$AUC = 0,8964$

Una vez ejecutado el sistema NLMS sin indicador de causa comuna, se puede observar a simple vista que existe una mejora existencial en comparación al sistema realizado con el filtro LMS, sin indicador de causa comuna. En esta ocasión se ha obtenido un área bajo la curva máxima $AUC = 0,9097$ en comparación al detector LMS sin indicador, ha aumentado el AUC un valor de 1,12 %. Aunque el Área Bajo la Curva es una unidad objetiva para evaluar el sistema, muy última, también hay que tener en cuenta el gráfico, pues en esta ocasión la Probabilidad de Falsa Alarma empieza a aumentar a una mayor Probabilidad de Detección.

A pesar haber conseguido mejora considerable, casi igualando el AUC resultante del detector LMS con indicador de causa comuna, se espera mejorar el sistema aún más, con el indicador de causa comuna.

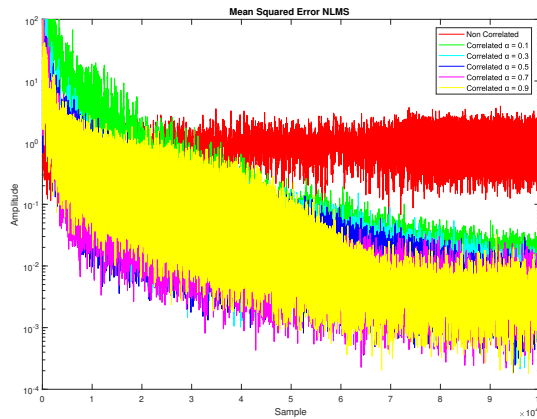


Figura 4.9: Error Cuadrático Medio a la salida del detector para detección con señales correlacionadas e incorrelacionadas

Es interesante observar a la salida del detector NLMS la diferencia entre el nivel del error cuadrático medio de una realización con señales incorrelacionadas frente el nivel del error cuadrático medio de una ejecución con

señales correlacionadas, en la figura 4.9. Se observa que con el uso de un sistema NLMS, cuando existe causa comuna el MSE a la salida del sistema es muy próximo a cero en comparación con el LMS.

Detección con indicador de causa comuna NLMS

El segundo escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante el siguiente indicador de causa comuna:

$$q[n+1] = (1 - \lambda)q[n] + \lambda e^2[n] \quad (4.34)$$

Con la presencia del indicador se pretende reducir la probabilidad de error de detección. Tal y como está definido, el indicador reduce los posibles picos que se producen en el error, y que podrían generar una falsa alarma.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.8: Parámetros iniciales para la detección NLMS con indicador de causa comuna

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Longitud señales de entrada:	100000 muestras
Controlador velocidad decrecimiento del indicador (λ):	0.002

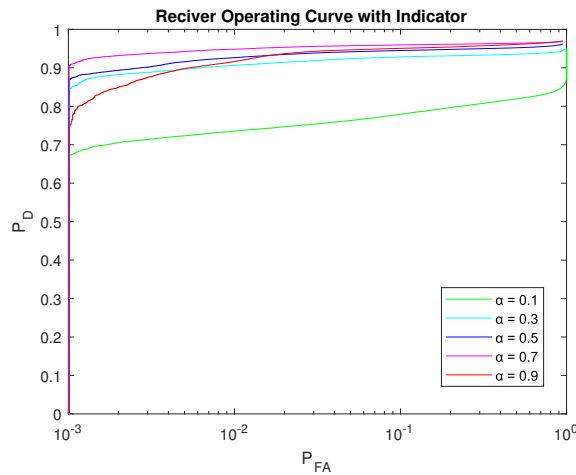


Figura 4.10: Característica Operativa del Receptor del detector NLMS con indicador de causa comuna

Cuadro 4.9: Resultados para los distintos valores de la variable α del detector NLMS sin indicador de causa comuna

Configuración	AUC
$\alpha = 0,1$	$AUC = 0,8140$
$\alpha = 0,3$	$AUC = 0,9171$
$\alpha = 0,5$	$AUC = 0,9026$
$\alpha = 0,7$	$AUC = 0,9141$
$\alpha = 0,9$	$AUC = 0,9048$

Por último, se ha ejecutado la simulación del detector NLMS con indicador de causa comuna. Al igual que sucedía anteriormente con el detector LMS, después de aplicar el indicador de causa comuna se espera obtener una mejora. En esta realización se ha conseguido un Área Bajo la Curva máxima $AUC = 0,9171$, con ello se ha logrado elaborar un sistema con exactitud alta, $AUC > 0,9$. Se trata de un muy buen resultado, pues el sistema tendrá una probabilidad de error muy baja.

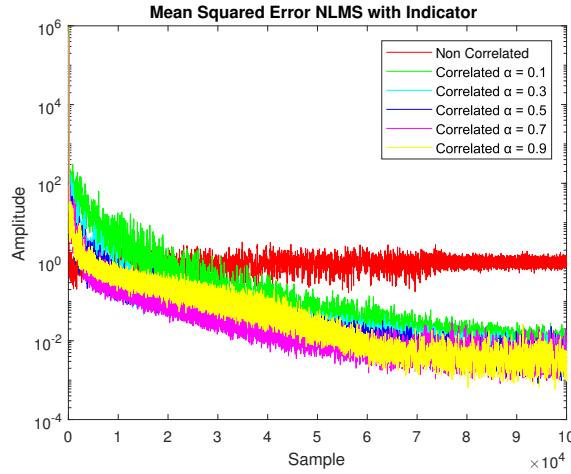


Figura 4.11: Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas

En la figura 4.11, se puede apreciar como la señal error cuadrático medio cuando las señales entrada del sistema son correlacionadas, decrecen esta convergen en un valor próximo a cero. A primera vista, se observa la diferencia entre la figura 4.11 y la figura 4.9, mientras en la realización sin indicador existen cambios bruscos en señal MSE que se usará para decidir si hay causa común o no, cuando se usa el indicador estos cambios se atenúan, gracias a la diferencia de los niveles el umbral de decisión puede ser más elevado.

4.4.3. Detector Affine Projection

Para elaborar el detector AP, se han usado dos filtros AP, definidos en la sección 3.6, configurándolos, como se expone en la subsección 4.2.1, de modo que a la salida del sistema se obtiene una señal error que será la clave para determinar si las señales entrantes están o no correlacionadas.

Con el objetivo de inspeccionar el comportamiento del sistema, se mostrarán sus ejecuciones en distintos escenarios:

Detección sin indicador de causa común Affine Projection

El primer escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante la señal error a la salida del detector AP.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.10: Parámetros iniciales para la detección AP sin indicador de causa común

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Tamaño matriz A y B:	$L \times N = 100 \times 2$
Longitud señales de entrada:	60000 muestras

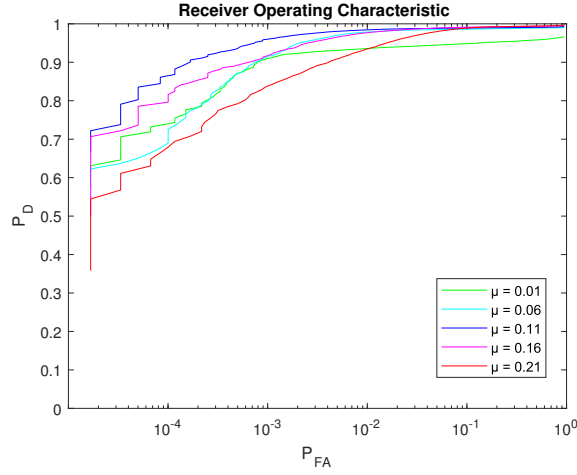


Figura 4.12: Característica Operativa del Receptor del detector AP sin indicador de causa comuna

Cuadro 4.11: Resultados para los distintos valores de la variable μ del detector AP sin indicador de causa comuna

Configuración	AUC
$\mu = 0,01$	$AUC = 0,9086$
$\mu = 0,06$	$AUC = 0,9384$
$\mu = 0,11$	$AUC = 0,9406$
$\mu = 0,16$	$AUC = 0,9429$
$\mu = 0,21$	$AUC = 0,9415$

Aunque no era el objetivo de la elección del algoritmo Affine Porjection para compararlo con el detector LMS y el NLMS, este ha logrado mejorar los resultados obtenidos en términos de la curva ROC respecto a los anteriores. Configurándolo con una tasa de aprendizaje $\mu = 0,16$ se ha obtenido un Área Bajo la Curva máxima $AUC = 0,9429$, lo que es un muy buen resultado.

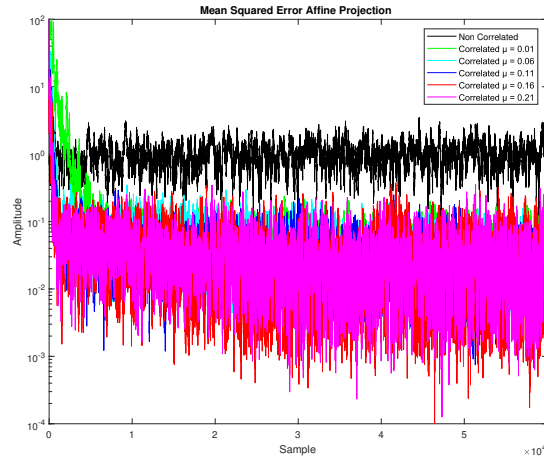


Figura 4.13: Error Cuadrático Medio a la salida del detector AP para detección con señales correlacionadas e incorrelacionadas

Como se puede observar en la figura 4.13, en comparación a las simulaciones realizadas con los filtros adaptativos LMS y NLMS, el tiempo de convergencia medio se ha reducido considerablemente tal que $t_{LMS} \approx t_{NLMS} \approx 3 \cdot t_{AP}$

Detección con indicador de causa comuna Affine Projection

El segundo escenario se pretende determinar la existencia de correlación entre las señales entrantes mediante el siguiente indicador de causa comuna:

$$q[n+1] = (1 - \lambda)q[n] + \lambda e^2[n] \quad (4.35)$$

Con la presencia del indicador se pretende reducir la probabilidad de error de detección. Tal y como está definido, el indicador reduce los posibles picos que se producen en el error, y que podrían generar una falsa alarma.

La ejecución se llevará a cabo con los siguientes parámetros:

Cuadro 4.12: Parámetros iniciales para la detección AP con indicador de causa comuna

Parámetro	Valor
Tamaño de los filtros:	100 coeficientes
Nº de ejecuciones para realizar la ROC:	100
Tamaño matriz A y B:	$LxN = 100x2$
Longitud señales de entrada:	60000 muestras
Controlador velocidad decrecimiento del indicador (λ):	0.002

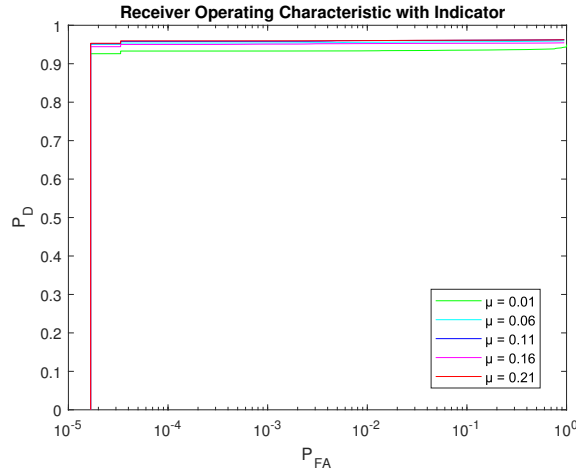


Figura 4.14: Característica Operativa del Receptor del detector AP con indicador de causa comuna

Cuadro 4.13: Resultados para los distintos valores de la variable μ del detector AP con indicador de causa comuna

Configuración	AUC
$\mu = 0,01$	$AUC = 0,9379$
$\mu = 0,06$	$AUC = 0,9095$
$\mu = 0,11$	$AUC = 0,9128$
$\mu = 0,16$	$AUC = 0,9049$
$\mu = 0,21$	$AUC = 0,9124$

En esta realización del experimento, se observa como el resultado respecto al uso del algoritmo sin indicador de causa comuna no ha mejorado. Al tratar-se de una diferencia de un 0,5 % su puede considerar que el efecto del indicador es nulo frente al sistema.

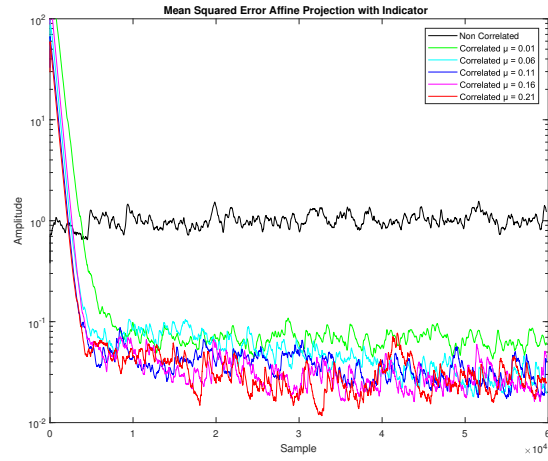


Figura 4.15: Error Cuadrático Medio con indicador a la salida del detector para detección con señales correlacionadas e incorrelacionadas

Para lo que si resulta muy útil el identificador es para identificar la muestra a partir de la que se puede considerar la correlación detectada. Mediante la figura 4.15 se puede determinar que de media el sistema si converge a partir de la muestra 6000.

Capítulo 5

Conclusiones

Recapitulando un poco, el estudio que se ha estado desarrollando tenía como objetivos identificar la existencia de una causa común entre dos señales aparentemente sin relación alguna. Al igual que muchas otras cosas, los resultados del descubrimiento de la existencia de un origen común entre dos procesos es aplicable a muchos campos de la ciencia, en estudios sociales, economía, física, medicina y otros, y para cada uno de ellos la finalidad podría ser totalmente distinta. Es muy interesante ver como algo que se ha estado estudiando a lo largo del grado, a priori era de uso exclusivo para telecomunicaciones, puede llegar a ser un concepto tan genérico que nos brinda a los estudiantes un abanico de posibilidades enorme para nuestro futuro.

Con el ojo puesto en el objetivo, se estableció una hoja de ruta para el proyecto que consistía primero de todo en identificar un sistema conociendo su señal input y su señal output. Para poder identificar el sistema mediante un filtro adaptativo, se debía configurar este de modo que la señal input estuviese configurada como los datos y la señal output como la señal deseada, después de cierto número de iteraciones, dependiendo del filtro usado, los coeficientes del filtro adaptativo serían los mismos que los coeficientes del filtro a identificar.

Existen una gran variedad de filtros adaptativos, muchos de ellos son de gran complejidad y con un coste computacional muy elevado. Para evitar complejidad de cálculo, primero se decidió elegir el filtro LMS, Least-Mean-Square, y su correspondiente normalizado el NLMS, Normalized-Least-Mean-Square, ya que estos no requieren realizar ninguna inversa matricial, algo que puede acarrear problemas en la ejecución y un gran coste computacional. Una vez dominada la identificación con los filtros adaptativos LMS y NLMS, se decidió tratar usar otro tipo de algoritmo que permitiese aumentar la velocidad de identificación, el algoritmo elegido fue el Affine Projection Algorithm. El Affine Projection es un algoritmo situado entre medio del LMS y de los algoritmos Recursive Least Square, RLS. Todos estos algoritmos trabajan en base al Least Mean Square Error, LMSE, y los coeficientes del filtro adaptativo son actualizados en cada iteración.

Cada uno de los algoritmos usado tiene sus ventajas y desventajas, además de sus peculiaridades a continuación, se van a comparar.

Para el algoritmo LMS, una de sus principales características es su simplicidad y su facilidad de cálculo, para el LMS no es necesario evaluar la inversa de la matriz de correlación a diferencia del algoritmo steepest descent, pero debido a su parámetro step size, μ , es fijo, es posible que el sistema se convierta en inestable debido a un sobredimensionado de la tasa de aprendizaje.

Por otro lado, tal y como se define el algoritmo NLMS, se soluciona el problema de la estabilidad del algoritmo LMS, pues en vez de usar una tasa de aprendizaje fija, se crea una tasa variable en cada iteración. Definiendo el parámetro step size variable también se logra reducir el tiempo de convergencia pero, en cambio, se incrementa el desajuste.

Por último, con el algoritmo Affine Projection se consigue reducir notablemente el tiempo de convergencia, es por ello que se seleccionó dicho algoritmo, a pesar de requerir realizar una inversa, el sistema tal y como esta definido permite hacer matrices mediante la señal entrante reduciendo así el número de iteraciones a realizar y por consiguiente el tiempo de convergencia, por contrapartida en este algoritmo el desajuste es mayor que en los anteriores.

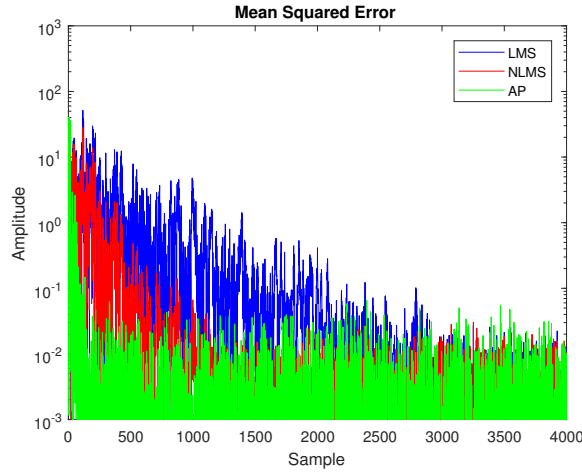


Figura 5.1: Error Cuadrático medio para cada algoritmo

En la figura 5.1, donde se muestra el error cuadrático medio para cada uno de los filtros, se puede observar la diferencia en el tiempo de convergencia de cada algoritmo. Como se ha comentado anteriormente la velocidad de convergencia del Affine Projection es superior a la del algoritmo NLMS y a la del LMS. El gráfico muestra como el Affine Projection es 30 veces más rápido que el LMS y aproximadamente 10 veces más rápido que el NLMS, lo que es una mejora más que considerable en tiempo de convergencia.

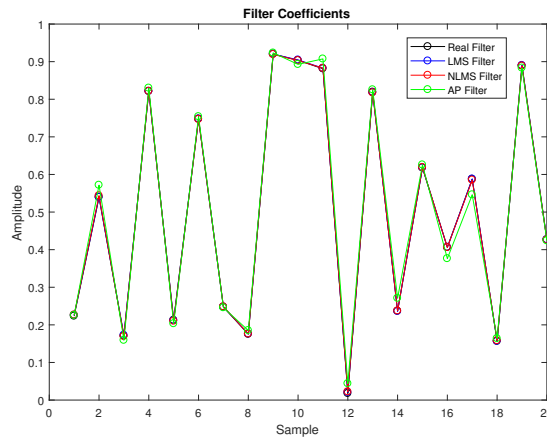


Figura 5.2: Coeficientes de los filtros

Por otro lado la figura 5.2, se puede observar lo comentado en relación al desajuste de los coeficientes identificados, en el caso del algoritmo LMS el desajuste es muy pequeño, como máximo del orden de 10^{-3} . Para la simulación con el algoritmo NLMS el desajuste es muy similar, aunque superior al del LMS, como máximo del orden también del 10^{-3} . Por último, para el algoritmo Affine Projection, se observa que el desajuste puede llegar a ser 10 veces mayor que para los dos anteriores, como máximo del orden de 10^{-2} .

Con los resultados obtenidos, se llega a la conclusión de antes de elegir el filtro con el que se va a identificar el sistema hay que definir bien las prioridades que se tienen, dependiendo de la aplicación. Está claro que si la prioridad de la identificación es la velocidad, sin duda el sistema a elegir sería el Affine Projection, en cambio, si la prioridad es la precisión, la mejor opción sería el algoritmo NLMS, pues este no muestra problemas con estabilidad y con un desajuste muy similar al del LMS, su velocidad es superior.

Una vez llegados a este punto no hay que olvidar el objetivo principal del trabajo, la detección de causa común en señales con aparente origen totalmente distinto. Para conseguir dicha detección se ha procedido a acoplado dos identificadores de sistemas y la amplitud de la señal error resultante será la señal que se elegida para tomar la decisión.

El modo de proceder ha sido el de usar los sistemas ya implementados en el capítulo 3 y acoplándolos, entonces realizar distintas simulaciones con el objetivo de ver su comportamiento para el caso en que existe una causa

comuna y para el caso opuesto. Además con el objetivo de aumentar la probabilidad de detección del filtro se ha decidido aplicar un indicador para evitar los cambios de valor bruscos en el MSE.

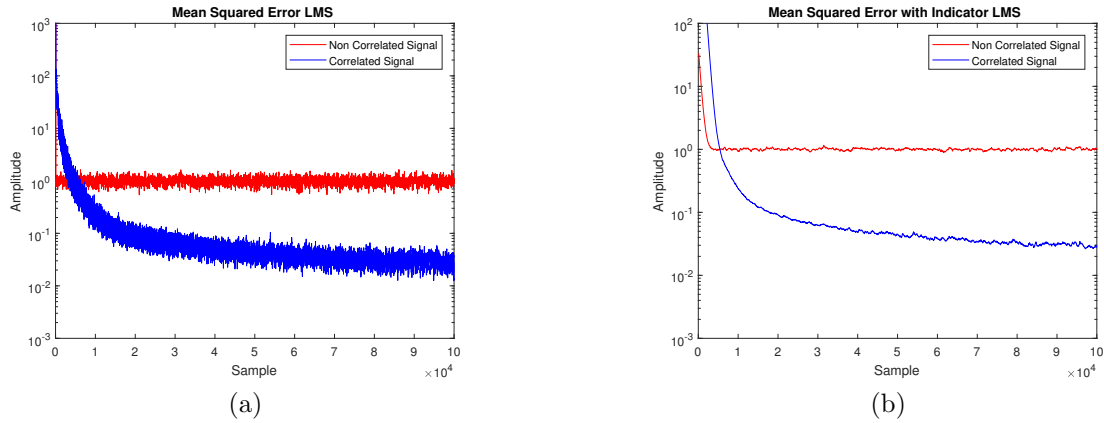


Figura 5.3: (a) Error Cuadrático Medio sin indicador y (b) Error Cuadrático Medio con indicador

Como se puede observar en la figura 5.3, la decisión va a ser más sencilla con el uso del indicador de causa comuna, pues si las oscilaciones son de gran amplitud, estas podrían generar una decisión errónea.

Los gráficos del MSE resultante pueden ser muy útiles para ver a simple vista si los resultados obtenidos son coherentes, pero para juzgar el detector de un modo objetivo el mejor modo ha sido usar la curva ROC, ya que esta evalúa el sistema y nos permite obtener un valor comprendido entre 0 y 1, siendo 1 el caso óptimo, para determinar si el detector es bueno o no.

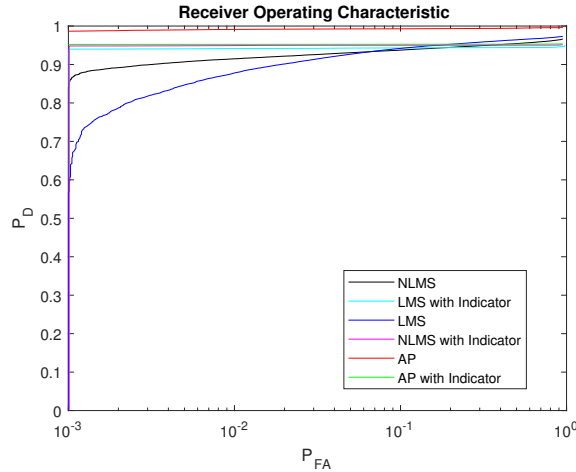


Figura 5.4: Curva Operativa del Receptor para cada detector

Como se puede observar en la figura 5.4 el mejor de los detectores es el AP sin indicador de causa comuna, con un Área Bajo la Curva tal que $AUC = 0,9429$, un resultado muy positivo, pues se encuentra en la categoría de filtros con alta exactitud. Además, la Probabilidad de Falsa Alarma del detector empieza a aumentar a partir de un valor considerablemente alto de la Probabilidad de detección. Esto no significa que los filtros restantes no sean buenos, pero si verifican que debido a la posible inestabilidad del filtro LMS este no obtenga resultados tan positivos como el AP. Además, debemos hacer especial hincapié en que el filtro AP es capaz de realizar la detección en un tercio de iteraciones que el filtro LMS y NLMS cosa que aún lo hace más atractivo y cumple las expectativas que teníamos frente a él.

Llegados a este punto y para concluir con el trabajo, se puede decidir que el mejor de los detectores ha sido el filtro AP sin indicador de causa comuna debido a que este eliminaba los posibles problemas de inestabilidad del LMS debido a la elección de la tasa de aprendizaje y ofrecía mejor velocidad de convergencia que el LMS y el NLMS, que en algunas ocasiones es lo más importante, como en este proyecto que se le ha dado mucha importancia. Por último, el indicador ha dado un punto de mejora a todos los filtros ya que al evitar los cambios bruscos en el MSE resultante, se consigue una señal más suave que sigue la misma tendencia que el MSE sin indicador.

Apéndice A

Identificador de Sistemas mediante LMS

```
1 N=1000000;
2 x = randn(1,N+2);
3 h = rand(1,500);
4 L = length(h);
5 y = conv(x,h);
6 Px = x*(x');
7 mu = 8/Px;
8 inicial = 0;
9 coeact = inicial*ones(L,1);
10 coe = zeros(L,N);
11 error_re = zeros(1,N);
12 xvec = zeros(L,1);
13 for n = L+1:N
14     xvec = x(n-1:-1:n-L)';
15     error_re(n) = y(n-1)-coeact'*xvec;
16     coeact = coeact+mu*xvec*error_re(n);
17     coe(:,n) = coeact;
18 end
```

Apéndice B

Identificador de Sistemas mediante NLMS

```
1 N=100000;  
2 time = linspace(1,N,N);  
3 x = randn(1,N+2);  
4 h = rand(100,1);  
5 L=length(h);  
6 y = conv(x,h);  
7 mu = zeros(1,N);  
8 alpha = 0.005;  
9 delta = 0.01;  
10 inicial = 0;  
11 coeact = inicial*ones(L,1);  
12 coe = zeros(L,N);  
13 error_re = zeros(1,N);  
14 xvec=zeros(L,1);  
15 for n=L+1:N  
16     xvec=x(n-1:-1:n-L)';  
17     Rx=xvec'*xvec;  
18     mu(n) = 2*alpha/(delta+((xvec')*xvec));  
19     error_re(n)=y(n-1)-coeact'*xvec;  
20     coeact=coeact+mu(n)*xvec*error_re(n);  
21     coe(:,n)=coeact;  
22 end
```

Apéndice C

Identificador de Sistemas mediante Affine Projection

```
1  L=10;
2  N=5;
3  G=1000;
4  mu=0.04;
5  x = randn(1,G+2);
6  h = randn(1,L);
7  y = conv(x,h);
8  y=y+0.05*randn(1,length(y));
9  inicial = 1;
10 Niter=1;
11 error_re_acu=0;
12 error_re_sq_acu=0;
13 coeact_final=0;
14 i=0;
15 for iter=1:Niter
16   wlact = inicial*ones(L,1);
17   wl = zeros(L,G-N-L+1);
18   e = zeros(N,G-N-L+1);
19   e_total=0;
20   for n=N+L:G
21     dn = y(n-1:-1:n-N);
22     dn = dn';
23     A = zeros(L,N);
24     for k=1:N
25       A(:,k) = x(n-k:-1:n-k-L+1)';
26     end
27     A=A';
28     e(:,n-N-L+1)=dn-A*wlact;
29     be=A'*(A*A')^-1;
30     wlact=wlact+mu*(A'*(A*A')^-1)*e(:,n-N-L+1);
31     wl(:,n-N-L+1)=wlact;
32   end
33   for n=1:G-N-L+1
34     if n == 1
35       e_total= e(:,n)';
36     else
37       e_total=[e_total e(:,n)'];
38     end
39   end
40   error_re_acu=error_re_acu+e_total;
41   error_re_sq=e_total.^2;
42   error_re_sq_acu=error_re_sq_acu+error_re_sq;
43   coeact_final=coeact_final+wlact;
44   i=i+1
45 end
```

Apéndice D

Detector de Correlación LMS

```
1 function [abs_error_re] = detector_lms (L, p, x, y, mu, N)
2 inicial = 10;
3 coeact_u = inicial*ones(L,1);
4 coeact_v = inicial*ones(L,1);
5 coeact_u_p = inicial*ones(L,1);
6 coeact_v_p = inicial*ones(L,1);
7 coe = zeros(L,N);
8 coe_u = zeros(L,N);
9 coe_v = zeros(L,N);
10 error_re = zeros(1,N);
11 xvec=zeros(L,1);
12 for n=L+1:N
13 xvec=x(n-1:-1:n-L)';
14 yvec=y(n-1:-1:n-L)';
15 error_re(n)=coeact_u'*xvec-coeact_v'*yvec;
16 coeact_u_p=coeact_u_p-mu*xvec*error_re(n);
17 coeact_v_p=coeact_v_p+mu*yvec*error_re(n);
18 alpha = 1/(sqrt(coeact_u_p'*coeact_u_p+coeact_v_p'*coeact_v_p));
19 coeact_u = coeact_u_p*alpha;
20 coeact_v = coeact_v_p*alpha;
21 coe_u(:,n)=coeact_u;
22 coe_v(:,n)=coeact_v;
23 end
24 abs_error_re = abs(error_re);
25 end
```

Apéndice E

Detector de Correlación NLMS

```
1 function [abs_error_re] = detector_nlms (L, p, x, y, N)
2 mu_x = zeros(1,N);
3 mu_y = zeros(1,N);
4 alpha_m = 0.7;
5 delta = 0.01;
6 inicial = 10;
7 coeact_u = inicial*ones(L,1);
8 coeact_v = inicial*ones(L,1);
9 coeact_u_p = inicial*ones(L,1);
10 coeact_v_p = inicial*ones(L,1);
11 coe = zeros(L,N);
12 coe_u = zeros(L,N);
13 coe_v = zeros(L,N);
14 error_re = zeros(1,N);
15 xvec=zeros(L,1);
16 for n=L+1:N
17 xvec=x(n-1:-1:n-L)';
18 yvec=y(n-1:-1:n-L)';
19 mu_x(n) = 2*alpha_m/(delta+((xvec')*xvec));
20 mu_y(n) = 2*alpha_m/(delta+((yvec')*yvec));
21 error_re(n)=coeact_u'*xvec-coeact_v'*yvec;
22 coeact_u_p=coeact_u_p-mu_x(n)*xvec*error_re(n);
23 coeact_v_p=coeact_v_p+mu_y(n)*yvec*error_re(n);
24 alpha = 1/(sqrt(coeact_u_p'*coeact_u_p+coeact_v_p'*coeact_v_p));
25 coeact_u = coeact_u_p*alpha;
26 coeact_v = coeact_v_p*alpha;
27 coe_u(:,n)=coeact_u;
28 coe_v(:,n)=coeact_v;
29 end
30 abs_error_re = abs(error_re);
31 end
```


Apéndice F

Detector de Correlación Affine Projection

```
1  inicial= 1;
2  L=100;
3  N=2;
4  mu=0.05;
5  G=30000;
6  wlact_u = inicial*ones(L,1);
7  wlact_v = inicial*ones(L,1);
8  wlact_u_p = inicial*ones(L,1);
9  wlact_v_p = inicial*ones(L,1);
10 y = randn(1,G+2);
11 x = conv(y,ones(1,30));
12 e = zeros(N,G-N-L+1);
13 e_total=0;
14 for n=N+L:G
15   A = zeros(L,N);
16   B = zeros(L,N);
17   for k=1:N
18     A(:,k) = x(n-k:-1:n-k-L+1)';
19     B(:,k) = y(n-k:-1:n-k-L+1)';
20   end
21   A=A';
22   B=B';
23   e(:,n-N-L+1)=A*wlact_u_p-B*wlact_v_p;
24   wlact_u_p=wlact_u_p-mu*(A'*(A*A')^-1)*e(:,n-N-L+1);
25   wlact_v_p=wlact_v_p+mu*(B'*(B*B')^-1)*e(:,n-N-L+1);
26   alpha = 1/(sqrt(wlact_u_p'*wlact_u_p+wlact_v_p'*wlact_v_p));
27   wlact_u_p = wlact_u_p*alpha;
28   wlact_v_p = wlact_v_p*alpha;
29   end
30   for n=1:G-N-L+1
31     if n == 1
32       e_total= e(:,n)';
33     else
34       e_total=[e_total e(:,n)'];
35     end
36   end
```

Apéndice G

Indicador de Causa Comuna

```
1 function [q] = indicador (lambda ,abs_error_re)
2 q = zeros(1,length(abs_error_re));
3 for n=1:length(abs_error_re)-1
4 q(n+1)=(1-lambda)*q(n)+lambda*abs_error_re(n)^2;
5 end
6 end
```

Bibliografía

- [1] Bernard Widrow, John McCool, and Michael Ball. The Complex LMS Algorithm. *Proceedings of the IEEE*, 63(4):719–720, 1975.
- [2] Jaume Riba Segarra. *Procesado de Señal Audiovisual y de Comunicaciones*. Barcelona, 2017.
- [3] Sajjad Ahmed Ghauri and Muhammad Farhan Sohail. System identification using LMS, NLMS and RLS. *Proceeding - 2013 IEEE Student Conference on Research and Development, SCORED 2013*, (December):65–69, 2015.
- [4] Alberto Gonzalez, Miguel Ferrer, Felix Albu, and Maria De Diego. Affine projection algorithms: Evolution to smart and fast algorithms and applications. *European Signal Processing Conference*, (Eusipco 2012):1965–1969, 2012.
- [5] Hyun-chool Shin, Ali H Sayed, and Woo-jin Song. Variable Step-Size NLMS and Affine Projection Algorithms. 11(2):132–135, 2004.
- [6] Galen Andrew, Raman Arora, Jeff Bilmes, and Karen Livescu. Deep Canonical Correlation Analysis. *Proceedings of The 30th International Conference on Machine Learning*, 28:1247–1255, 2013.
- [7] Ignacio Santamar, Javier Vía, and Jesús Pérez. A Robust RLS Algorithm for Adaptative Canonical Correlation Analysis. *IEEE Signal Processing Magazine*, pages 365–368, 2005.
- [8] W. W Peterson, T. G Birdsall, and W. Fox. The theory of signal detectability. *Information Theory, Transactions of the IRE Professional Group on*, 4(4):171–212, 1954.
- [9] Cristiano Leite Castro and Antonio Padua Braga. Optimization of the Area under the ROC Curve. *2008 10th Brazilian Symposium on Neural Networks*, pages 141–146, 2008.

